Exemplifying smart functions for a next generation data analytics toolbox

Abou Eddahab, F.

DOI
10.4233/uuid:ec128e53-e78a-4550-8aa5-0fafb36a7763

Publication date
2020

Document Version
Final published version

Citation (APA)
Abou Eddahab, F. (2020). *Exemplifying smart functions for a next generation data analytics toolbox*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:ec128e53-e78a-4550-8aa5-0fafb36a7763

Exemplifying smart functions for a next generation data analytics toolbox

Dissertation

For the purpose of obtaining the degree of doctor

at Delft University of technology

by the authority of the Rector Magnificus Prof. dr. T.H.J.J van der Hagen;

Chair of the Board of Doctorates

To be defended publicly on

Wednesday, 21 October 2020 at 10:00 o'clock

by

Fatima-Zahra ABOU EDDAHAB

Master of Research in Industrial Engineering

Grenoble University of Technology, Grenoble, France

Master of Engineering in Mechanical Engineering

Mohammadia School of Engineers, Rabat, Morocco

born in Meknes, Morocco

This dissertation has been approved by the promotor:

Prof. dr. I. Horváth

Composition of the doctoral committee:

Rector Magnificus	chair person
Prof. dr. I. Horváth	Delft University of Technology, promotor

Independent members:

Prof. dr. Z. Simeu-Abazi	Grenoble Institute of Technology, France
Prof. dr. D. Kiritsis	Swiss Federal Institute of Technology Lausanne, Switzerland
Prof. dr. E. du Bois	University of Antwerp, Belgium
Prof. dr. P. Lucas	Twente University, the Netherlands
Prof. dr. ir. S. Sariyildiz	Delft University of Technology, the Netherlands
Prof. dr. ir. F.E.H.M. Smulders	Delft University of Technology, the Netherlands

Reserve member:

Prof. dr. J. Schoormans	Delft University of Technology, the Netherlands

This Ph.D. research was funded by the European Union.

Exemplifying smart functions for a next generation data analytics toolbox

Keywords: Data analytics; middle-of-life data; white goods designer; data analytics toolbox; user identification; data streams merging; recommender system; axiomatic theory fusion

Ph.D. thesis
Delft University of technology, Delft, The Netherlands
ISBN: 978-94-6384-162-7
An electronic version of this dissertation is available at http://repository.tudelft.nl/

Acknowledgements

"*Man will not get anything unless he works hard*" – Surah al-Najm, 53:39.

The completion of this Ph.D. thesis would not have been possible without God's help and the efforts of several people to whom I am extremely grateful. First and foremost, I would like to thank and express my gratitude to my Promotor Professor Imre Horváth for believing in me and giving me the opportunity to join his team and work on this exciting research project. He has been not only my promotor, but also my daily supervisor, my mentor and a colleague with whom I had fruitful conversations and nice debates about science, technology, cultures and life in general. His dedication to his job and his passion for technological advances and the important role of research made me proud to be working with him and taught me to enjoy hard work and research challenges. I discovered with him that staying in my comfort zone will never make me a good researcher. Thank you Professor for believing in me all along this journey, for making me the researcher I am today and for being the example to follow in my career.

I would like to thank Dr. Wilfred van der Vegte for guiding me in the beginning of my Ph.D. project and for introducing me to several aspects of the Dutch culture. I will always remember our good discussions and amazing project trips. In the same occasion, I would like to thank Dr. Zoltan Rusák for his advices and work consultation during the Ph.D. period. I learned a lot from his positivity, the way of approaching research problems as well as his out of the box thinking. Special thanks go out to the outstanding committee members of my dissertation, who assessed and approved my thesis, Professor Zineb Simeu-Abazi, Professor Dimitris Kiritsis, Professor Els du Bois, Professor Peter Lucas, Professor Sevil Sariyildiz, Professor Frido Smulders and Professor Jan Schoormans. Thank you for allocating time and efforts to review my work and to be present in my Ph.D. defense. Thank you for your guidance, remarks and contributions to improve the quality of my thesis report.

My Ph.D. journey would not have been enjoyable without my wonderful teammates and colleagues with whom I had so much fun in the last four years. Special thanks to Sirasak, Yongzhe, Shahab and Santiago for their encouragements and help. It was a privilege sharing the office with you. I also would like to thank the secretaries of the Design Engineering department at the Delft University of Technology for their support and help through all aspects. Thank you Sara, Jolanda, Mariska and Selina for making my TU Delft experience as smooth as possible.

Most importantly I would like to thank my family for their unconditional love and support. To my mom and dad, Najia and Dr. Hassan, thank you for making me the person I am today. Thank you for pushing me to follow my dreams even if that meant being far from you. Thank you for your limitless emotional and financial supports that allowed me to achieve my objectives in life. I can never thank you enough for the way you raised me, the freedom you gave me and the peaceful and healthy atmosphere you created and maintained in our house. You have always been and you will always be my role models and my pride. I am lucky and grateful to have you as my parents. To my sisters, Dr. Zineb and Dr. Chaimae, thank you for being always there for me. Thank you for our laughs and your constant encouragements. You helped me keep going to be a good example of an elder sister. Your successes are always my motivation and my source of happiness when things get difficult.

To my husband Steven, thank you for appearing at the right moment in my life and helping me finish my Ph.D. in the most productive way. Thank you for believing in me when my frustrations took over. Constantly repeating that “You are smart. You can do this.” made me overcome my worries and insecurities. You took my role, the wife’s role, to make me focus on my research without me asking for it and without you complaining about it. Your love, your understanding, your patience and your presence balanced my life in the most perfect way. To my in-law family, Irene, Sherwin and Bryan, thank you for being my family abroad. Our talks and excursions were all I needed to get rid of my stress and recharge my batteries. Thank you Irene for opening your house to me and sharing all about your trips and work experience and how to keep being positive and love your job in all circumstances. Our long discussions taught me life lessons.

This acknowledgement could not be complete without dedicated this work to the memory of my aunt Amina, that left us last year. Thank you for all the love and affection you gave me and my sisters. You will remain in my heart until the end of time.

Table of contents

CHAPTER 3: BUILDING A QUALITATIVE THEORY AND FRAMEWORK OF THE NEEDS 51

CHAPTER 4: CONCEPTUALIZATION OF A DEMONSTRATIVE SMART DATA ANALYTICS TOOLBOX 111

CHAPTER 5: IMPLEMENTATION AND VALIDATION OF THE FUNCTIONS OF THE DEMONSTRATIVE SMART DATA ANALYTICS TOOLBOX 155

List of figures

Chapter 1:

Chapter 2:

Chapter 3:

Chapter 4:

Chapter 5

List of tables

Chapter 2:

Chapter 3:

Chapter 5:

List of acronyms

ATF	Axiomatic Theory Fusion
ANN	Artificial Neural Network
API	Application Programming Interface
AR	Auxiliary Requirement
BDA	Big Data Analytics
BDH	Big Data Handling
BDP	Big Data Processing
BoL	Beginning-of-Life
BR	Basic Requirement
C	Cluster
CBR	Case-Based Reasoning
Cl	Commercial
CNN	Conventional Neural Network
CSD	Color Structure Descriptor
CWF	Computational workflow
DAT	Data Analytics Tool
DC	Data Category
DIR	Design Inclusive Research
DL	Deep Learning
DML	Deep Machine Learning
DNN	Deep Neural Network
DO	Output of Data Analytics
DS	Data Source

EER	Extension of Entity-Relationship
EHD	Edge Histogram Descriptor
EoL	End-of-Life
ER	Entity-Relationship
EU	European Union
FMEA	Failure Modes and Effects Analysis
G-EHD	Global Edge Histogram Descriptor
GUI	Graphical User Interface
HMM	Hidden Markov Models
HMMD	Hue-Max-Min-Diff
I	Impact
ICA	Independent Component Analysis
IR	Interface Requirement
IoT	Internet of Things
k-NN	k-Nearest Neighbor
LBPH	Local Binary Pattern Histogram
LDA	Linear Discriminant Analysis
L-EHD	Local Edge Histogram Descriptor
MBR	Memory Based Reasoning
MED	Minimum Expected Difference
ML	Machine Learning
MoL	Middle-of-Life
MoLD	Middle-of-Life Data
MoLD-S	Middle-of-Life Data Stream
MSDM	Multi-Sensor Data Merging
N	Need

ODR	Operative Design Research
OO	Object-Oriented
OS	Open-Source
PCA	Principal Component Analysis
PP	Processing Performance
Q	Question
RC	Research Cycle
RDC	Research in Design Context
S	Storage
SDATB	Smart Data Analytics Toolbox
SGD	Stochastic Gradient Descent
SG-EHD	Semi-Global Edge Histogram Descriptor
SVM	Support Vector Machine
Symrcm	Symmetric Reverse Cuthill-McKee Reordering
UI	User Interface
QBI	Web-hosted Questionnaire-Based Interrogation
WRQ	Working Research Question

Chapter 1

Introduction

1.1. Falcon project

The presented research was part of a European Union (EU) funded project called "Feedback mechanisms across the lifecycle for customer-driven optimization on innovative product-service design," referred to as "H2020 – Falcon." FALCON explored using manufacturing intelligence to support innovative product-services design. It focused on customer satisfaction and the future efficiency of companies and aimed at deploying user experiences and user data collected via the Internet of things (IoT) and social media to improve product-service systems. The project included several academic and industrial participants, such as Philips Healthcare and Arçelik. Its goal was to provide new forms of connections and interactions between users, products, services, digital devices, and their dynamic environment to extend the entire lifecycle of product-service combinations. The project had several objectives: (i) addressing product-service information collection through collaborative intelligence and product-embedded information devices; (ii) enabling product-service knowledge representation, exploitation, openness, and diffusion; (iii) strengthening collaboration and new product-service development through new feedback and feed forward mechanisms; (iv) supporting innovative product-services design using manufacturing intelligence; and (v) improving product-service lifecycle assessment approaches.

The Falcon project was intended to deliver significant impact for EU citizens and industry at different levels. The expected potential impacts consisted of the following:

- Increased market knowledge, enabled by the continuous collection of product experiences, which will foster the development of new product-services tailored to the expectations of social groups;
- New business models, developed through the analysis of user feedback and benchmarking of other markets;
- Innovation, enabled by product-embedded information devices and context awareness for self-improvement throughout the whole product lifecycle;
- Cost-effective products, enabled by selective simplification of products and waste reduction;
- Process efficiency, enabled by collaborative tools that allow product, service, and process designers to learn and understand how networked intelligent products in the IoT can be an advantage;
- Enhanced serviceability, through the establishment of directions to develop proximity added services and thus European employment; and

- Business growth, by ensuring successful adoption of new products and services through improvements and better anticipation of consumer expectations.

From this perspective, the project participants were supposed to provide a framework to enable the realization of new products and value-adding services by monitoring the real use of products and services in operation to extend product and service lifespans and to optimize the use of the necessary resources all along their lifecycle.

1.2. Setting the stage

The rapid rise of emerging information and knowledge economies and the deployment of information technologies have led to remarkable changes in the lifecycle of products and services. Because of the fast-growing informatization of the development of hardware and software products, the focus on exploring data has become ubiquitous [1]. Product development organizations are extracting data to glean insights into information patterns that will subsequently feed back into the product development process. Companies need to adjust their operations in response to the influences of rapidly evolving markets and to better manage the lifecycles of their products. To this end, efforts are devoted to combining (i) static process information with dynamic information, (ii) product information including process and resource information, and (iii) human aspect information with business information throughout the entire product lifecycle [2]. The implementation of the concept of "run-time" data-driven design proved to be an effective facilitator, as the ultimate goal of companies producing consumer durable goods is to maintain their competitiveness over the longest possible period of time [3].

Data about the use of products and services can provide useful insights and facilitate design enhancements. However, given the changes in data (i.e. shifting from small to big data), effective data analytics needs dedicated analysis, simulation, and forecasting tools. These changes have raised new challenges for computational processing. That is why a new form of data science is emerging, and numerous methods and tools have been developed in the field of data science and engineering. The recent developments in data analytics and the application of data analytics tools have opened a new path for generating knowledge for product [4]. Accordingly, product developers can achieve perpetual enhancement of their products and services based on real-life use, work, and failure data.

It is useful to see how the use of products by different customers can provide insights companies can employ to transform use patterns to design enhancements based on data generated from those products. This data can be accessed and collected from product sensors, log files, or web resources (social media, forums, etc.). The potential knowledge gained from analyzing data can help reduce project time, improve product quality, and increase customers' satisfaction [5]. It fosters organizational actions and help firms establish sustainable competitive advantage [6]. It can also support strategic design decisions and, consequently, boost and create competitive advantages. Despite the efforts to develop data analytics tools, the same attention is not paid to all phases of the product lifecycle.

Most previous efforts were dedicated to the methodological and computational support of beginning-of-life (BoL) and end-of-life (EoL) models and activities. Few efforts were made to exploit middle-of-life (MoL) data and activities and to create knowledge and value from this type of data. Thanks to new information technologies (sensors, smart tags, etc.), the chunks of information conveyed during the MoL phase of products can finally be identified, tracked, and collected [7]. Unfortunately, analyzing and feeding MoL data and use patterns to product designers remains an insufficiently addressed issue [8]. Considering all the elements mentioned above, there seems to be a lack of dedicated data analytics tools and techniques to support product enhancement using MoL data (MoLD).

1.3. Research phenomenon

We are witnessing the era of smart products [9]. Today, these products are able to sense, learn, store, and share information about their use and users [10]. This progress overwhelms product designers with tremendous amounts of generated data, and traditional data analytics tools are incompatible for scaling to big data [11]. The outdated performance of traditional data analytics tools makes them unable to manage and extract practical knowledge from big data [12]. Neither are existing data analytics tools tailored to deal with specific data exploitation situations, such as supporting the enhancement of products by designers. Smartly aiding designers is still a superficially explored domain, although it offers many new opportunities. This is the broad phenomenon that was the motivation for this research, and the fundamental assumption was that innovative, efficient, interoperable, and scalable big data analytics solutions are needed to analyze big data obtained from diverse sources [13].

Tailoring data analytics and knowledge-mining tools for adequately processing large datasets has become a necessity [14]. This is especially true when the intent is to switch from BoL and EoL to MoL data analysis. Most of the existing tools were developed to process BoL and EoL data, whereas smart products can also generate MoL use data. The switch to MoLD is important and offers benefits for data processing, since MoLD generate opportunities to continuously evaluate and enhance products and services [15]. In other words, MoLD can be transformed into knowledge that can enable perpetual and long-term design improvement, product innovation, and product planning.

Data analytics tools (DATs) present several challenges, such as the following:

- Managing rapidly changing patterns of use and operational data;
- Dealing with generic DATs in specific product development cases;
- Combining tools from an information processing point of view to cover all data transformation steps;
- Combining and integrating the outcomes of various data analytics tools; and
- Interpreting the meanings of these outcomes in the context of the product development tasks at hand [16].

One more insufficiency from the perspective of the tools was reported: “The addition of environment and external data would demand that new analytics tools are developed to effectively identify and extract knowledge for making decisions in a design process” [17]. Although numerous data analytics (software) tools and packages have been developed for extracting product-associated data, the practice of exploiting data analytics methods and tools for product enhancement is still in a rather immature stage [18]. Many elaborations on related issues can be found in the literature, but no convincing solutions are yet included in commercialized data analytics systems.

1.4. Research goals

There is a lack of computational mechanisms to support decision-making and servicing, as well as a lack of theories explaining how to select, combine, and deploy existing mechanisms and software tools in the case of product-use data (MoLD). The overall objective of this thesis is to cover the lack of data analytics tools designers need to process MoLD. Towards this end, one of the goals of the Ph.D. research was to generate requirements and fundamentals for a new smart data analytics toolbox (SDATB) able to overcome the issues limiting existing tools and convert them into functionalities. Figure 1.1 sketches the research objectives: one is design practice oriented (implicit goal), and the other one is technology development oriented (explicit goal). They can be underpinned by the following argumentation.

The ultimate objective is to support designers in product enhancement based on MoLD. Effective statistical and semantic processing of MoLD is not only an academic challenge but also a useful asset for the industry [19]. It is important for product developers and production companies to learn how their products are used under different circumstances. This may provide insights on how to avoid deficiencies that may occur under circumstances that were not completely known or specified in the development phase of their products. MoLD can be aggregated by making field observations and interrogating users, or by studying failure log files and maintenance reports, or from relevant web resources. Alternatively, these data can be elicited directly from products by sensors or self-registrations.

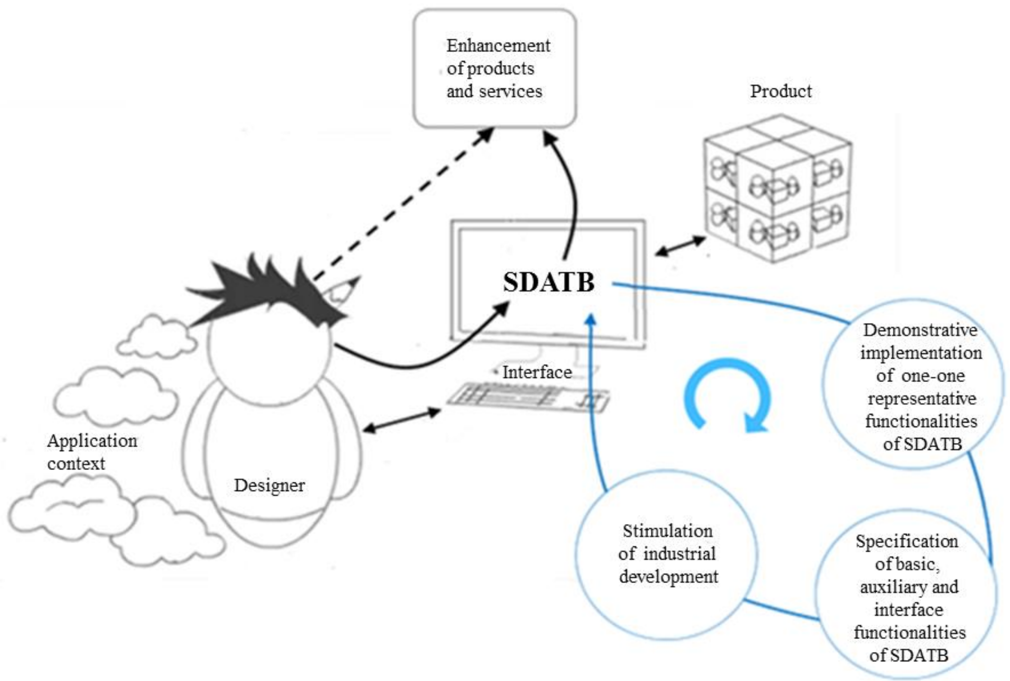


Figure 1.1. Research goals

The last mentioned approaches are becoming more popular as products advance from traditional free-standing products to network-linked advanced products to awareness- and reasoning-enabled smart products [20]. However, given the dynamic changes in sensor data, the large volumes of data aggregated over time, and the unknown nature of data patterns, it is unfortunately not straightforward to perform effective data analysis using existing traditional techniques [21]. Feeding structured MoLD back to product designers is an insufficiently addressed issue [8]. The key challenge is to find ways to use data analytics techniques in purposeful combinations effectively, based on the application contexts and the specific objectives of product designers [22].

Another possible and operationalized aim of the Ph.D. research is the development of demonstrative functional elements of a smart toolbox. The SDATB is seen as the next generation enabler for designers facing data analytics challenges. Obviously, due to the need for extensive research, programming, and testing, the development of the SDATB as a whole could not be targeted in the Ph.D. research project. In the thesis, only demonstrative technology development—that is, technology exemplifying certain functions of a SDATB—could be included. The three main milestones of the demonstrative technology development loop are shown in Figure 1.1. The explicit aim was to present examples of smart basic, auxiliary, and interface functions. These are elaborated on in the dissertation and are brought to an algorithmic implementation as one–one representative and demonstrative elements of the SDATB.

1.5. Research questions

The main guiding research question of this work has been formulated as follows:

> *What functionalities are to be included in a next generation smart data analytics toolbox to help product designers enhance products and services based on MoLD?*

To answer this question, three groups of working research questions (WRQs) have been identified. The first group concerns knowledge aggregation and the building of a knowledge platform and contains five WRQs:

- WRQ_1: What is the state of the art in the development of (smart) data analytics tools (or toolboxes) in the context of product enhancement?
- WRQ_2: What are the limitations of existing traditional data analytics tools?
- WRQ_3: What elements influence the development of data analytics tools in the context of product improvement by product designers?
- WRQ_4: Why is smartness needed to develop a next generation SDATB?
- WRQ_5: What requirements should be considered for the elaboration of the SDATB?

The second group concerns the conceptualization of the demonstrative SDATB and contains four WRQs:

- WRQ_6: In what way can the requirements be converted into functionalities for the SDATB?
- WRQ_7: What functionalities are to be provided by the SDATB?

- WRQ_8: What are examples of basic, auxiliary, and interface functions to be included in a demonstrative SDATB?
- WRQ_9: What are the considerations on which the example basic, auxiliary, and interface functions can be realized?

The third group concerns the implementation and the validation of the demonstrative SDATB and contains five WRQs:

- WRQ_{10}: What algorithms and data constructs are needed for the implementation of the sample basic, auxiliary, and interface functions?
- WRQ_{11}: In what way can the sample basic, auxiliary, and interface functions be implemented?
- WRQ_{12}: How can the usefulness of the sample basic, auxiliary, and interface functions be demonstrated?
- WRQ_{13}: What support services can be expected from an all-embracing computational implementation of the proposed functions of an SDATB?
- WRQ_{14}: What novelty does the proposed SDATB present from academic and industrial points of view?

The above questions are answered in the upcoming chapters of this thesis, based on the methodology presented in the next section.

1.6. Methodological framing of the research

The Ph.D. work was divided into four research cycles (RCs) and was designed based on the framing of three methodologies presented in [23]. These are (i) research in design context (RDC), (ii) design inclusive research (DIR), and (iii) operative design research (ODR). RDC supports analytical disciplinary research aiming at insights, understanding, and predictions. The research relied mainly on the knowledge of background disciplines. It used many research methods of these disciplines and lent itself to monodisciplinary approaches. RDC also concentrates on building and providing theories, which add to the disciplinary knowledge of design. DIR supports constructive disciplinary and operative design research by involving various manifestations of design in research as research means. It integrates knowledge of multiple source domains and lends itself to multidisciplinary insights, explanations, and predictions. This methodology generates knowledge, “know how,” and tools for problem-solving. In general, ODR extracts knowledge from concrete practical design processes, environments, and artifacts. It supports the improvement of design problem-solving intelligence reflexively and offers generally valid principles, rules, and standards. In this research, ODR was related to practical testing of the implemented demonstrative functions.

The methodological framing of the research used the principles mentioned above. It helped in summarizing and harmonizing the initial plans for the research content and processes. This framing facilitated the transformation of the theoretical framework of the SDATB into a testable prototype. The research cycles, their designs, and their logical flow are illustrated in Figure 1.2.

RC_1 was framed according to RDC methodology. It addressed WRQ_1 – WRQ_4 and was dedicated to overviewing and evaluating the state of the art related to existing DATs to support product and service enhancement using MoLD. The explorative part explored four main domains of interest: (i) the nature of data, (ii) data transformation approaches, (iii) data analytics tools and packages, and (iv) applications of data analytics. The confirmative part consisted of synthesizing the findings and building a robust knowledge platform about availabilities, limitations, and opportunities related to SDATB development.

Similarly, RC_2 was framed according to RDC methodology. It addressed WRQ_4 and WRQ_5. To achieve the objective of defining requirements for the SDATB, two main activities were carried out: an inductive study and a deductive study. The explanatory phase of the inductive study consisted of two research actions: (i) a web-hosted questionnaire-based interrogation (QBI) and (ii) a literature study. The former was intended to investigate designers' needs for new knowledge from a practical point of

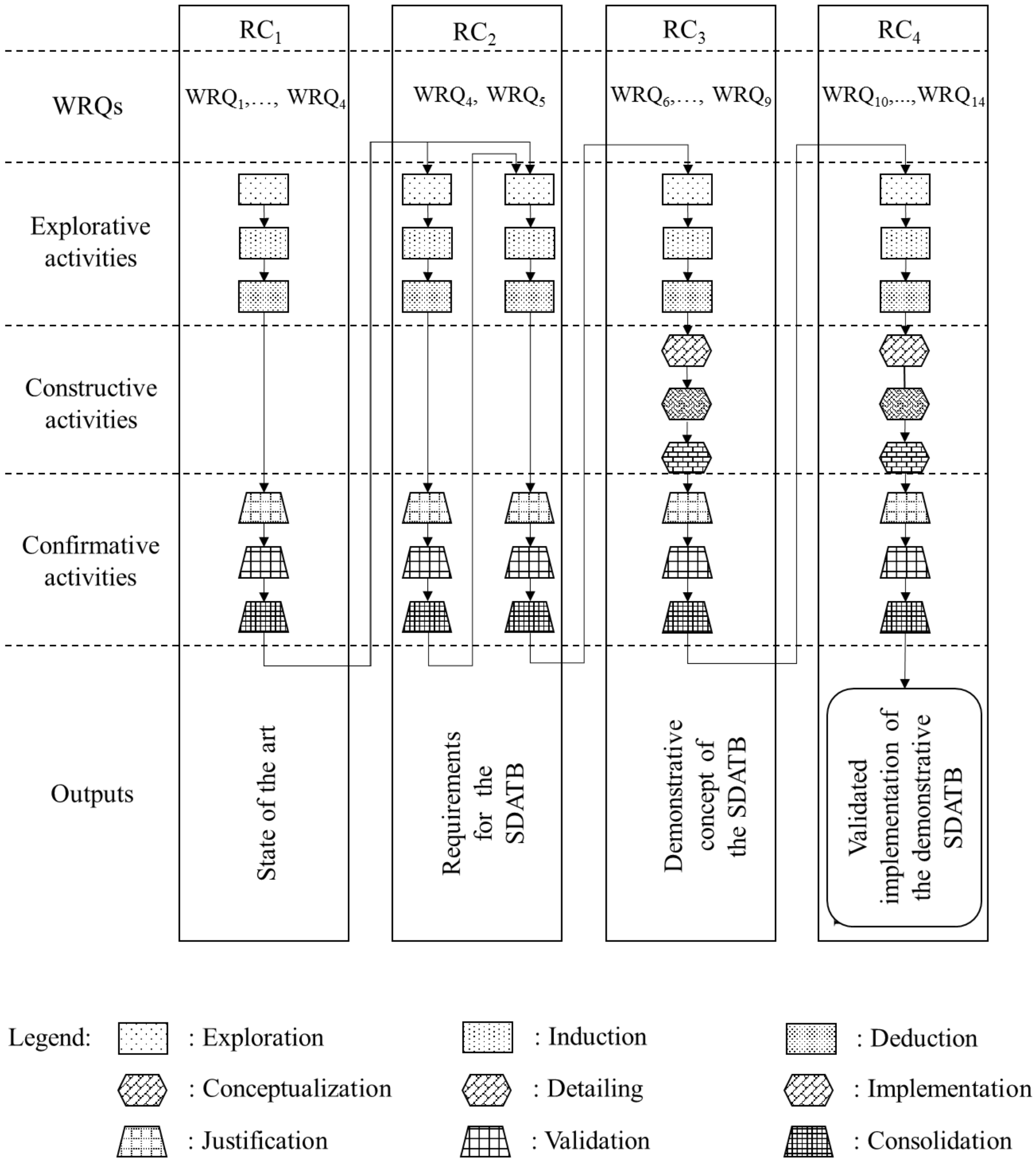


Figure 1.2. Methodological framing of the overall research

view, while the latter was done to derive knowledge from a state-of-the art point of view. The outcomes of both activities were synthesized into theories. The validation phase of the inductive study compared the two obtained theories to identify their congruent and incongruent parts. Based on this, the theories could be complemented and consolidated. The deductive study involved axiomatization-based conceptual discretization of relevant theories and semantic fusion of the axioms and the supplementary postulates into the body of a new and synthetic explanatory theory. The exploratory phase used the outcomes of the inductive study and complemented them with an investigation of existing theories relevant for building data analytics tools. The outcomes of this phase where used in the confirmation phase, in which a new theory consolidating fundamentals, requirements, and expected functionalities of the SDATB was generated.

RC_3 was framed according to DIR methodology. It answered WRQ_6 – WRQ_9. This research cycle conceptualized a demonstrative SDATB. The explorative part aggregated knowledge from previous research cycles (fundamentals, requirements, etc.) that served as the bases for ideation, selection of the most relevant ideas, and exploration of existing enabling technologies. This served the purpose of specifying the fundamental concepts related to an SDATB and the functionalities of a next generation data analytics toolbox. The constructive part of this research cycle focused on filtering the obtained toolbox functionalities to establish a comprehensive conceptualization of a demonstrative SDATB. Towards this end, the conceptualization and architecture of functionalities chosen for implementation were produced after the investigation of concept methods from a computational point of view.

RC_4 was formulated partly according to the DIR methodology, partly according to the ODR methodology. It sought to find answers to WRQ_{10} – WRQ_{14}. This research cycle was dedicated to the implementation and validation of the demonstrative SDATB and its components related to functional, architectural, and algorithmic considerations. The explorative phase of this research cycle collected and sorted information on the prototype-level implementation of demonstrative SDATB functionalities and the executable algorithms and computational techniques. The constructive phase of the research cycle focused on software-level implementation of all functionalities and algorithms of the demonstrative SDATB. Finally, the confirmation phase tested and validated the feasibility and performance of the executable algorithms and the interpretation of their results. A function evaluation scenario was generated for the validation of the three implemented (basic, auxiliary, and interface) demonstrative functions.

1.7. Thesis outline

The overall methodological framing presented in Figure 1.2 was used to organize the overall activities of this thesis. The research cycles and their concrete research actions are specified and detailed successively in the upcoming chapters. Chapter 2 provides an overview of the literature study we conducted. This study investigated affordances in the context of data analytics tools as well as the conceived limitations present in the state of the art. In addition, all elements and domains influencing the development of DATs were investigated. The findings were summarized and used in building a knowledge platform that is used as a basis for the rest of the RCs.

The outcomes of the literature study conducted in Chapter 2 formed the starting point of Chapter 3, dedicated to investigating concrete, practical designers' needs in the context of the study via a QBI and confronted to the literature to produce a complete image of what is missing in DATs that would satisfy product designers. A complementary study consisted of building a new theory based on designers' needs but also an investigation of theories needed for DATs development. The methodology used for combining all relevant theories is called axiomatic theory fusion (ATF), and it is applied in the concrete application case of product designers using MoLD to enhance white goods. The details of the methodology, its components, and its processes are also presented in this chapter. The expected outcomes of Chapter 3 are a set of fundamentals, requirements, and functionalities needed for the SDATB composition. In Chapter 4, the fundamentals, requirements, and functionalities of the SDATB are summarized and filtered for the conceptualization of a demonstrative concept of the toolbox. The functions chosen for the toolbox are articulated and decomposed to the lowest level of functions (elementary functions) to facilitate the definition of the algorithms needed for the computational implementation.

In Chapter 5, the algorithms and data constructs needed for the realization of the demonstrative functionality of the smart toolbox are specified and detailed. They together form a part of the computational mechanisms of the SDATB. An application case as defined, and the representative computational functions are tested in the context of this application. Implemented as interoperating algorithms and data constructs, the representative basic, auxiliary, and interface functions of the SDATB are validated for their performance. In Chapter 6, the complete research project is summarized to answer the main research question of this scientific project: What functionalities are to be included in a next generation smart data analytics toolbox to help designers enhance products and services based on MoLD? This chapter is a reflection on all research activities conducted in the four research cycles and their findings. This reflection is formulated in terms of conclusions, propositions, and recommendations for future research.

1.8. Related own publications

1. Abou Eddahab, F.-Z., & Horváth, I. (2018). What do designers miss regarding the outputs of data analytics tools in the context of possible product improvements? In: *Proceedings of the 12th Symposium on Tools and Methods of Competitive Engineering*, 1-14.
2. Abou Eddahab, F.-Z., & Horváth, I. (2018). What does data analytics offer for extracting knowledge from middle-of-Life product data? In: *Proceedings of the 25th International Conference on Transdisciplinary Engineering,* 7, 1102-1111.
3. Abou Eddahab, F.-Z., & Horváth, I. (2019). Using data analytics to extract knowledge from middle-of-life product data. *International Journal of Advanced Research and Publications,* x-x.
4. Abou Eddahab, F.-Z., & Horváth, I. (2020). Recommendation function for smart data analytics toolbox to support choosing task-relevant data analytics tools. In: *Proceedings of the 13th Symposium on Tools and Methods of Competitive*

Engineering, x-x.

5. Abou Eddahab, F.-Z., & Horváth, I. (2020). Constructing a new theory based on axiomatic fusion of qualitative engineering theories. *Journal of Engineering design*, x-x.
6. Abou Eddahab, F.-Z, & Horváth, I. (2020). Recommendation function for smart data analytics toolbox to support semantic merging of middle-of-life data streams. In: *Proceedings of the 4th International Conference on Control, Automation and Diagnosis*, x-x.
7. Abou Eddahab, F. -Z, & Horváth, I. (2020). Two computational mechanisms for a smart data analytics toolbox: Merging middle-of-life data streams and choosing proper machine learning tools. (under preparation)

1.9. References

[1]. Bi, Z., & Cochran, D. (2014). Big data analytics with applications. *Journal of Management Analytics*, 1(4), 249-265.

[2]. Bodenhoefer, K., Schneider, A., Cock, T., Brooks, A., Sands, G., Allman, L., ... & Catanese, B. (2004). Environmental life cycle information management and acquisition–first experiences and results from field trials. In: *Proceedings of Electronics Goes Green*, 5-8.

[3]. Shapiro, N. (2005). Competition and aggregate demand. *Journal of Post Keynesian Economics*, 27(3), 541-549.

[4]. Li, Y., Thomas, M.A., & Osei-Bryson, K. M. (2016). A snail shell process model for knowledge discovery via data analytics. *Decision Support Systems*, 91, 1-12.

[5]. Rajpathak, T., & Narsingpurkar, A. (2013). Managing knowledge from Big Data analytics in product development. *Tata Consulting*, 11.

[6]. Sena, J. and Shani, A.B. (2000), Knowledge management and new product development: learning from a software development firm. In: *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management*, 19, 1-5.

[7]. Jun, H.B., Kiritsis, D., & Xirouchakis, P. (2007). Research issues on closed-loop PLM. *Computers in Industry*, 58(8-9), 855-868.

[8]. Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management-from its history to its new role. *International Journal of Product Lifecycle Management*, 4(4), 360-389.

[9]. Sankhe, P., & Rodrigues, E. (2018). Smart Backpack. In: *Proceedings of the Third International Conference for Convergence in Technology*, IEEE, 1-4.

[10]. Khan, W.Z., Aalsalem, M.Y., & Khan, M.K. (2018). Five acts of consumer behavior: A potential security and privacy threat to Internet of Things. In: *Proceedings of the International Conference on Consumer Electronics*, IEEE, 1-3.

[11]. Bhatnagar, R. (2018). Machine learning and big data processing: A technological perspective and review. In: *Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications*, Springer, 468-478.

[12]. Dhaenens, C., & Jourdan, L. (2019). Metaheuristics for data mining. *4OR*, 17(2), 115-139.

[13]. Rizwan, A., Zoha, A., Zhang, R., Ahmad, W., Arshad, K., Ali, N.A., ... & Abbasi, Q. H. (2018). A review on the role of nano-communication in future healthcare systems: A big data analytics perspective. *IEEE Access*, 6, 41903-41920.

[14]. Chou, S.C., Yang, C.T., Jiang, F.C., & Chang, C.H. (2018). The implementation of a data-accessing platform built from big data warehouse of electric loads. In: *Proceedings of the 42nd Annual Computer Software and Applications Conference*, IEEE, 2, 87-92.

[15]. Ericson, A., Müller, P., Larsson, T., & Stark, R. (2009). Product-service systems–from customer needs to requirements in early development phases. In: *Proceedings of the 1st CIRP Industrial Product Service Systems Conference*, 62-67.

[16]. Li, J., Tao, F., Cheng, Y., & Zhao, L. (2015). Big data in product lifecycle management. *The International Journal of Advanced Manufacturing Technology*, 81(1-4), 667-684.

[17]. Lewis, K., & van Horn, D. (2013). Design analytics in consumer product design: A simulated study. In: *Proceedings of the International Design Engineering Technical Conferences*. American Society of Mechanical Engineers. V03BT03A003-V03BT03A003.

[18]. Baraka, Z. (2014). Opportunities to manage big data efficiently and effectively. Doctoral dissertation, Dublin Business School.

[19]. Franke, M., Klein, P., Schröder, L., & Thoben, K.-D. (2011). Ontological semantics of standards and PLM repositories in the product development phase. *Global Product Development*. Springer, 473-482.

[20]. Bongard-Blanchy, K., & Bouchard, C. (2014). Dimensions of user experience-from the product design perspective. *Journal d'Interaction Personne-Système*, 3(1), 1-15.

[21]. Katal, A., Wazid, M., & Goudar, R. (2013). Big data: issues, challenges, tools and good practices. In: *Proceedings of the Sixth International Conference on Contemporary Computing*, IEEE, 404-409.

[22]. Van der Vegte, W.F. (2016). Taking advantage of data generated by products: trends, opportunities and challenges. In: *Proceedings of the ASME International 2016 International Design Engineering Technical Conferences*, Charlotte, USA, x-x.

[23]. Horváth, I. (2007). Comparison of three methodological approaches of design research. In: *Proceedings of the International Conference on Engineering Design*, 7, 28-31.

Chapter 2

Research cycle 1: Establishing a knowledge platform for investigation of data analytics technologies

2.1. Introduction

2.1.1. Objectives and activities of the first research cycle

The first research cycle aimed at building a knowledge platform concerning data analytics tools and packages. The overall objectives of this chapter were the refinement of the phenomenon, the research objectives, and the critical exploration of the state of the art of DATs. In this sense, the relevant knowledge domains were specified to find the gaps related to the research phenomenon and to landmark a direction for this research. The knowledge aggregation consisted of investigating DATs and all domains of interest that might have an influence of the development of DATs. The purpose was to analyze existing approaches and identify limitations of existing data analytics solutions and to develop a clear vision of what was missing in order to construct our own approach.

To formulate a descriptive theory identifying the boundaries and knowledge gaps, a substantial number of subscription-based and open access journals, conference proceedings, web repositories, and professional publications covering our domains of interest were studied carefully. This helped determine what could be addressed in research and what the open opportunities were for creation of new knowledge. In the orientation phase of the Ph.D. work, we observed that serious gaps exist related to data analytics computer support in the context of product enhancement by product designers using MoLD. Consequently, we identified four domains of interest and studied them to determine the current situation and to seek for opportunities for developing novel data processing technologies.

During the literature study, we found many useful sources in some fields, while other fields were weakly covered. This may be interpreted as an indication that research is still in its infancy in these fields. Examples include research related to (i) the smartness of data analytics tools, (ii) the usage of tools by practical designers who are not data specialists, and (iii) the MoLD usage in product enhancement. Some inconsistencies were also encountered in the literature study, namely the incorrect usage of some words and expressions as synonyms (e.g. smartness and intelligence or data processing, data analytics and data mining). Understandably, this issue led to a lack of clarity and a superficial understanding of problem. In our Ph.D. research, to avoid misunderstanding and misinterpretation, we tried to provide appropriate and expressive definitions of the concepts we used.

2.1.2. Methodology applied in the first research cycle

In the first research cycle, the applied methodological framing was RDC. Based on its principles, the activities of this cycle were conducted in two consecutive phases: explorative and confirmative. In the first phase, we (i) divided the research topic into several domains of interest, (ii) aggregated the knowledge of each domain, (iii) discussed the findings of the aggregation, and (iv) synthesized the findings to generate knowledge. In this exploratory phase, we organized the literature investigation into two sequential but interrelated steps. The first step was a shallow exploration that identifies the most relevant domains of knowledge for the study. Based on wide range of keywords a topographic landscape of related publications was developed. The second step was a deep exploration in which we collected several hundred relevant publications and intensively analyzed various sources of knowledge. In the second, confirmative phase of the research cycle, we analyzed the synthesized findings in the context of product enhancement, particularly with regard to the use of data analytics tools by designers to process MoLD. This analysis identified limitations of existing data analytics tools and packages. From these limitations, we identified opportunities for data analytics tools development.

2.1.3. Reasoning model of the literature study

We completed a comprehensive literature study in two phases. The first phase, referred to as shallow exploration, was conducted to identify the most relevant domains of knowledge for the study. Based on a wide range of keywords, we tried to develop a topographic landscape of the related publications. This topographic was meant to show not only the distribution of clusters of keyword-related publications but also the peaks and the plains of these clusters. Figure 2.1 shows the clustering resulting from keyword-

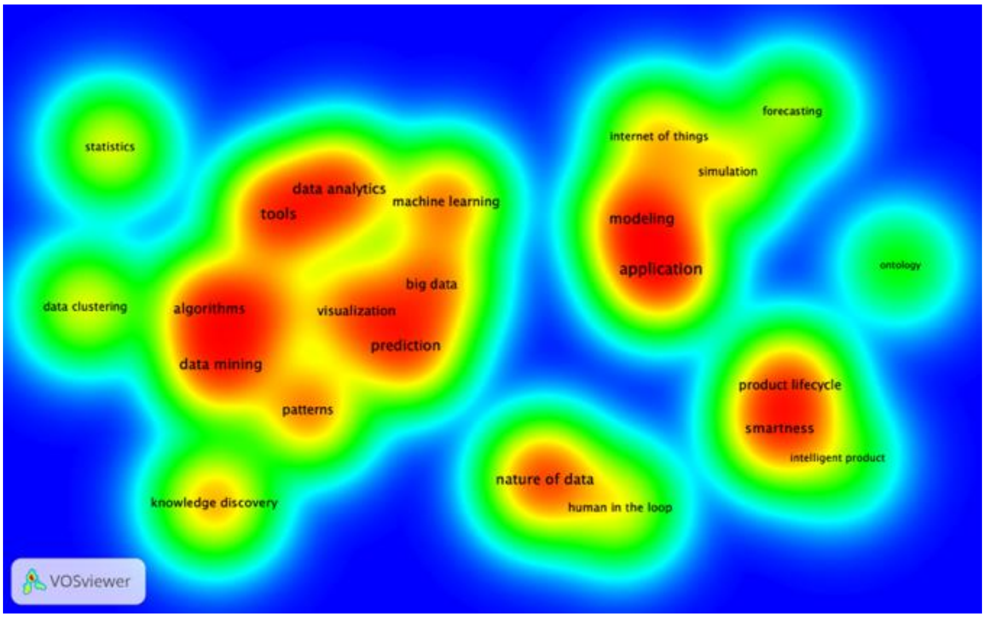


Figure 2.1. Occurrence of the chosen keywords in the literature

based mapping of the related literature. The graphical image was built using VOSviewer. Figure 2.1 shows not only the neighboring (semantically related) keywords but also the distances between them as they appear in the literature. The colors indicate the frequency of the occurrence of keywords (i.e. the formation of peaks). The most frequently occurring keywords are shown in red and dark orange, and the less frequent ones are shown in green and light blue. The visual representation generated by the software application let us recognize four major clusters of papers. In a kind of transitive ordering, these are as follows: (i) changes in the nature of data, (ii) approaches to transforming data, (iii) tools and packages for data analytics, and (iv) design applications for data analytics. These cluster labels were used as descriptors of the main domains of interest in the detailed literature study.

In the second phase of the literature study, called deep exploration, various sources such as subscription-based and open access journals, conference proceedings, web repositories, and professional publications were searched and several hundred relevant publications were collected. The findings made it possible to define further relevant key terms on a third level (not shown in Figure 2.1). The second phase was also used to quantitatively characterize the interrelationships among the key terms belonging to the same cluster. Figure 2.2 shows the interrelationships found. If two terms are used in the same document, then there is a line between them, and the thickness of the line indicates how frequently they occur. In other words, the thick lines refer to combinations of terms that appear in multiple papers, whereas the thin lines refer to combinations that rarely appear in the studied publications. The connectivity diagram in Figure 2.2 reveals that the thickest lines are between the above mentioned cluster labels– a fact that underlines their significance and relatedness. In addition, the diagram not only casts light on the complexity of the completed study but also indicates which key terms could not be studied separately because of how tightly they were interconnected in the studied publications.

The above information obtained from the quantitative part of the literature study were

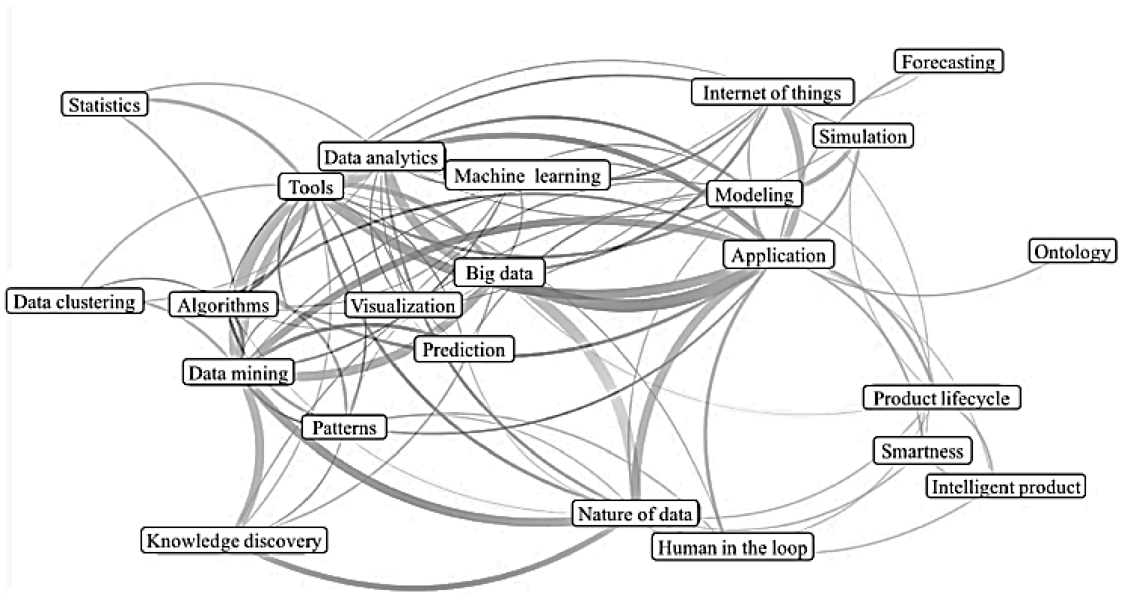


Figure 2.2. Connectivity graph of chosen keywords

used to develop a reasoning model for the qualitative part of the study. This part focused on interpreting the findings and disclosed semantic relationships. The reasoning model is shown in Figure 2.3. Only the first- and second-level key terms are indicated, whereas, as mentioned above, the study was actually done with key terms of the third decomposition level. The considered papers were published at different times, ranging from the mid-1950s until today. An important observation was that the concepts identified by the first-level key terms are in implicative relationships with each other. Specifically, if the nature of data changes, that entails a change in approaches to data transformation, which in turns implies the need for different data transformation methods and tools. These enablers can provide support for a broader range of existing applications and can facilitate new data analytics applications to enable product enhancement and innovation by design. The investigation into the changes in the nature of data was focused mainly on product-related use, maintenance, and service data and on data describing the conditions and behavior of products.

The next sections of this chapter review the state of the art in the broad field of data analytics methods and tools, which support extracting product developmental knowledge from MoL product data. First, we investigate the essence and trend of changes from product-associated data (referred to as functional data, or small data in other publications) to big data. Then, we review the various data transformation actions and techniques and discuss the accompanying challenges. Furthermore, we summarize our findings about existing commercial and academic data analytics (software) tools, and discuss how they can be improved according to the literature. Various applications of data analytics are also discussed, including the major application domains of various big data analytics approaches and the challenges that have already been recognized and

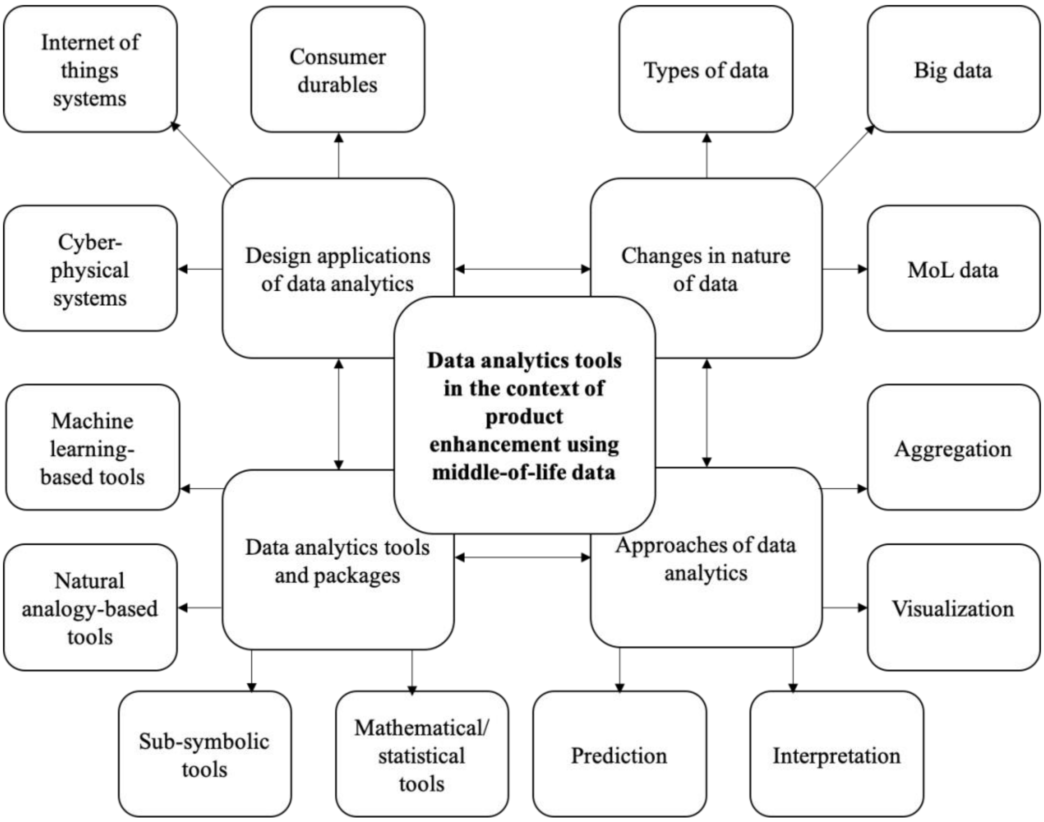


Figure 2.3. Reasoning model of the literature study

addressed. Finally, we combine the findings in the above four domains and discuss their implications.

2.2. Knowledge aggregation about the changes in the nature of data

2.2.1. Overview of the changes in the nature of data

Analysis of the trends and changes in the nature of data implies the need for a historical perspective. As a result of the third industrial revolution, which established the scientific and technological domain of electronics and involved it in automation of industrialized processes beginning in the 1950s, in addition to alphanumeric data, analogue system signals have been also carriers of "data." The fourth industrial revolution, which culminated in the mid-1980s, introduced digital computing and syntactic data processing not only in industrial contexts but also in everyday creative and executive processes. The current fifth industrial revolution, often referred to as the revolution of intelligence, shifted the attention to various formal and tacit forms of knowledge and to knowledge engineering and semantic knowledge processing. This is an indispensable step considering the objective of present-day product design and production to offer smart, cognizant, and even intelligent artifactual systems for society. These are deemed the most fundamental generic and global trends of change that can be identified in the nature of data. In addition to these, however, the literature also reflects many specific and local changes in the nature of data capture and processing, in particular in the field of the development of new products and value-adding services.

In the above context, data is a set of qualitative and/or quantitative values of variables [1]. It may concern the behavior, the status, or the function of a system and can be in different formats (symbols, texts, numbers, figures, etc.) [2] [3]. Before and at the beginning of the fourth industrial revolution, data were typically limited (up to a couple of petabytes, as a maximum) in terms of their quantity, indicated today by the term "small data." Due to technological advancements in digital data processing and the use of multiple digital devices and complex sensor and effector networks, the possibilities for capturing and processing data have drastically changed. Tremendous amounts of digital data records are generated, forming continuous data streams. This new generation of data, also called "big data," may run up to exabyte scales [4]. No longer are data regarded as static or stable but as dynamic and recomposable. As such, big data are raw material for business exploitation and a crucial input for creating new forms of economic assets and values [5]. The motivation to exploit mass data has emerged as a new research field called big data science. Big data analytics (BDA) has emerged as a promising and rapidly proliferating methodology to retrieve knowledge from massive data streams and repositories [6].

The overall objective of this section initially was to find and analyze scientific and professional publications that discussed the recent changes and trends in the nature of data. However, given the abundance of data types, the review actually conducted was restricted to data associated with monitoring real-life use of products and services in operation and to data obtained from user feedback on social media. This scoping of the study made it possible to derive highly relevant conclusions in the narrow context of our research. The practical objective was formulated to study the kind of data that makes it possible (i) to extend product and service lifespans and (ii) to optimize the use of necessary resources throughout their lifecycle. Here, the term "lifecycle" refers to all phases of the product and service life, from the BoL phase, when the product is designed and realized, through the MoL phase, when the product is available on the market and used by the customer, until the EoL phase, when the product is discontinued or revamped [7].

In addition to the enormous changes in terms of the amount of elicited and processed data, the move from natural, concrete, and unstructured data to (purposefully) created, abstract, and highly organized data structures is the most important change. This is both the outcome of and the stimulant for advanced database technologies. As discussed in [8], abstract data types can play a significant role in software development. Different data models and data modeling techniques, which are central to information systems, have been proposed that provide a basis for specific technological solutions for database design and for realizing data-intensive applications [9].

Data modeling is the kernel activity in data management and processing that imposes constructs and structures on data and elicits meaning from individual or structured data. Technically, data modeling is the process of (i) discretizing physical variations of data for accurate representation, (ii) modeling sets of data as data structures, and (iii) generating data constructs for effective processing by available software [10]. Data modeling is the basis of all data processing tasks and forms an explicit part of many of them, supporting the capture and understanding of the relationships and meaning of data [11]. Data modeling creates a simplified structure and representation of data that can be used as the starting point for analytics, reasoning, and simulation [12]. Data modeling also supports the schemata design of databases and data warehouses and repositories [13]. The logical structures imposed by data modeling support human understanding, maintaining and extending data structures [14]. Data models represent the structures and integrity of the elements of data [15]. Their semantics usually constitute an informal agreement between developers and users of data models [16]. As databases became critical components of information systems, the success of projects became dependent on the accuracy of data models [17] [18]. The huge amount of data has made data modeling a critical and vital issue of survival of companies [19].

Over the past decades of digital data processing, the fundamental concepts and general principles of data modeling have undergone an evolution [20]. There are seven modern data modeling approaches identified in the literature: (i) relational, (ii) semantic, (iii) entity-relationship (ER), (iv) extensions of ER (EER), (v) object-oriented (OO), (vi) statistical, and (vii) data metamodeling [21] - [23]. Although many pitfalls have been discovered in relational theory, relational data models have become widely accepted. They offer mathematical foundations and simple user-level paradigms, while semantic models offer flexible structuring capabilities and explicit data constraints [18]. Semantic modeling applies various abstractions (such as classification, instantiation, aggregation, decomposition, generalization, and specialization) to capture meanings in data structures in complex situations. Semantic integrity is typically guaranteed by making use of the fundamental type-attribute relationship. OO data modeling captures conceptual entities as objects [14]. OO modeling has emerged as an alternative to the traditional entity-relationship modeling technique, based on the premise that the resulting OO data models are easier to use and understand [24]. Experience has shown that OO data models can indeed be more quickly understood than EER models for both simple and complex problems. However, many of the other claims concerning the superiority of OO data modeling (e.g. its perceived ease of use) are not verified empirically. OO data models are typically used in computing – for instance, in programming, analysis, design, and database management [25]. Despite the huge number of research projects in semantic and OO data modeling, the relational database is still the predominant one used in the industry [26]. ER and EER modeling are popular as tools for conceptual data modeling [27]. For the purpose of exploratory data analysis (of continuous data), various parametric and nonparametric data models have been proposed based on estimating the quantile functions and density quantile functions [28] [29].

The trends of change concern not only the sources and amount (size) of digital data, but also the arrangement (structure) of data (see Figure 2.4). Unstructured data do not support formal analyses, traditional database management [30], or the application of pattern searching methods [31]. The papers related to the nature of data clusters indicate that individuals, industry, and science face the challenge of dealing with large datasets. This is a result of the proliferation and ubiquity of high-throughput computing technologies and internet connectivity. The main difficulty is not in the technical handling of large amounts of data but in mining and extracting valuable information and knowledge from them [32]. Decades ago, data were characterized by three characteristics (volume, velocity, and variety) because these lend themselves to advanced, complex, and predictive business analysis and insights [33] [34]. Recently these have been complemented with three more characteristics: value, veracity, and viability [35] [36]. To deal with big data's scalability and affordability, the literature suggests, requires optimized data warehouses and cloud computing [37]. The literature review made clear that managing and gaining insights from the produced big data is a challenge and a key to competitive advantage [38]. It offers substantial value to organizations who decided to adopt it, but poses a number of challenges to the realization of such benefit [39].

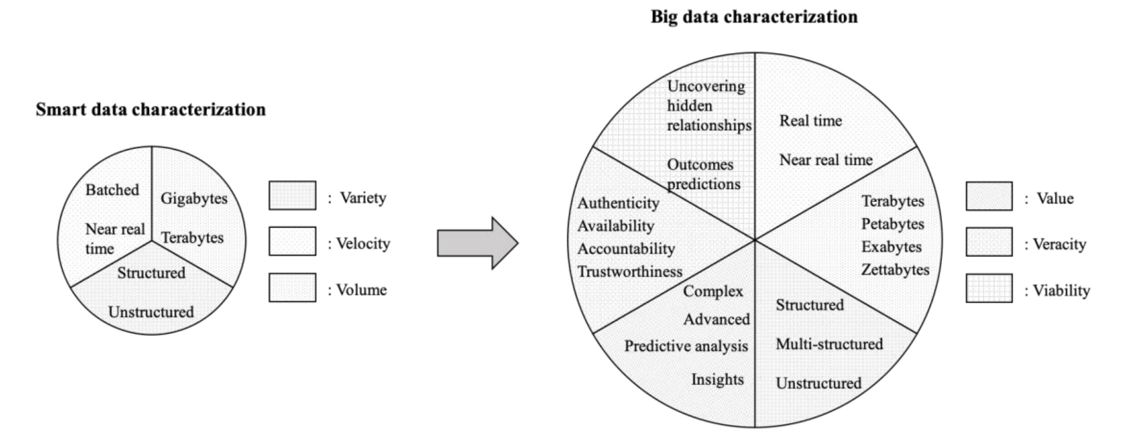


Figure 2.4. Change in the characteristics of regular and massive datasets (designed after [35] and [36])

Big data is collected during all product lifecycle stages: BoL, MoL, and EoL [7]. Researchers showed that it is important to focus on MoLD, which primarily, but not exclusively, includes use, service, and maintenance data [40] [41]. It includes failure data, performance data, product age data, operating environment data, usage intensity data, maintenance reports, and refund and replacement data [42]. These data allow observation of the conditions and behaviors of products during the usage phase [3]. If manufacturers face communication challenges and do not have well-established processes to obtain and use feedback from their customers, processing MoLD may also help them [43]. Furthermore, acquisition of MoLD creates opportunities for and encourages a lifecycle-oriented approach to incremental product design that evaluates and enhances all products and services on a continuous basis [44]. In other words, MoLD can be transformed into knowledge that enables perpetual design improvement and product innovation and planning. Collecting product information during the MoL and EoL phases allows for improvement of a product or product operations in various ways, such as design improvements and the optimization of maintenance operations [45].

A recognized difficulty related to MoLD is that the related elicitation activities should be executed outside the companies, typically with intense involvement of both the products and the end users. If elicitation of product-related information is interventional, it may lead to operational inefficiencies [40]. Since conventional information systems used in defining products and services cannot handle MoLD, the developers of product lifecycle management systems have recognized the need for dedicated data analytics approaches and tools. Nevertheless, the potentialities offered by MoLD analytics are seldom utilized by industrial product and service developers. The gradually increasing smart behavior of products has been recognized as a key development in collecting data and information on modes of use and operation and feeding it back to designers [40].

2.2.2. Lessons learned

The study of the literature showed that the notion of “data” and the nature of data, as well as the types of knowledge digitally processed, have changed rapidly and remarkably over a relatively short time. What is typical today is to produce and process complex data structures, rather than only data constructs or individual data, as contents for data warehouses and integrated databases [46]. We are moving away from the so called “situated aspect data” (sorted according to type, location, and meaning) to big data, which cannot, however, be interpreted easily by examining its structure and semantics. This is because characteristic and significant patterns are often deeply hidden in the flow of data, and nothing hints directly at the semantic meanings of the various bodies of data [47] [48]. One explanation for why digital data have become diverse is that engineering and technical data have been combined with social data [49]. At the same time, there is also a tendency to produce qualitative data together with quantitative data. This is now a daily routine. Processing lifetime data of products and related processes means that descriptive, prescriptive, predictive, and operational data should be managed concurrently. This represents a data engineering challenge for the developers of current data processing tools [50]. The need is growing to modify or redesign these tools to be able to reveal and process hidden data patterns and mixed semantics [51]. The above changes in the nature of data explain why a new data science is emerging and why many novel methods and tools have been or are being developed in the field of data management technologies.

Data modeling plays an important role in both structuring data and capturing and eliciting the meaning of (structured) data. Typically, three types of data models are used in information systems: (i) conceptual data models (data requirements models), (ii) logical data models (models of data constructs and structures), and (iii) physical data models (database or data warehouse access models) [52]. These models are used in both business-oriented and system-related data processing. The major issues are related to (i) the inflexibility of current data models, (ii) the multiplicity of imposing structures and capturing meanings, (iii) the limited reusability of data models, (iv) the insufficiency of data modeling standards, (v) the lack of stereotyped data interfaces, (vi) the management of the relational complexity of data models, and (vii) the uncoordinated development of information systems [53].

Although the above issues were identified almost 15 years ago, most are still acute. A clear distinction has been drawn between data modeling and data analytics. Data modeling is the activity of using a set of tools and techniques to aggregate, organize, relate, represent, and store data, whereas data analytics is interpreted as the activity of merging data from multiple sources using a set of methods and tools to gain insight from the data and analyze trends to help make better decisions [54]. An impenetrable range of tools for data modeling and representation and data analytics have been developed. This issue is addressed in Section 2.4.

At this point it is important to mention that, as with all data collected during the lifecycle of a product, MoLD also need to be defined, collected, and processed before they can be used. However, there is a difference concerning the execution of these four steps.

- MoLD are associated with certain phases of the product lifecycle. Therefore, first the phases MoLD are associated with should be captured. These phases can be as broad as logistics, operation, and maintenance, which produce a wide variety of MoLD, such as run-time performance data, failure data, data about the aging of a product, data on changes in the operating environment, usage intensity data, completed maintenance task data, and refund and replacement data. Due to their nature (state and time dependence), these data need specific data processing and interpretation approaches.
- Collecting data is determined where the MoLD can be found and collected. In this sense, MoLD may have many sources, the outcomes of which should typically be combined before processing. For instance, MoLD can be aggregated from field observations, interrogations of users, the study of failure log files and maintenance reports, or relevant web sources such as social media and user forums. Alternatively, they can also be elicited directly from products by sensors or self-registrations.
- Processing MoLD depends on the opportunities of computational management – for instance, on the preprocessing, processing, and post-processing procedures (detailed in the next section). Although these steps are common, what makes them challenging is the fact that the complexity of data should be addressed, the data is to be processed in real time, and there is a time- and context-dependence to be considered. Existing data processing tools are not yet equipped with capabilities for these purposes.

- The use of MoLD can be considered from multiple aspects and for multiple application cases. This poses different requirements for the output of data analytics tools. For instance, data collected during the use of a product may deliver insights about consumers' behavior and preferences but also about the frequency of product use, the convenience of the interaction, or the potential misuses of a product. This means that the goals of data processing are strongly articulated by the objectives in using a product and the manners in which it is used.

2.3. Knowledge aggregation about steps and techniques of data transformation

2.3.1. Mapping of data transformation approaches

The term "data transformation" has a broad meaning and a narrower meaning. In the narrower meaning, it is the process of converting data and information from one format to another, usually from the format of a source system into the required format of a destination system [55]. The typical statistical transformations include (i) logarithmic, (ii) square root, (iii) square, (iv) cube root, and (v) reciprocal transformations. In the broader meaning, data transformation refers to all data processing activities that can introduce change in the state, representation, and/or meaning of data [56]. That is, data transformation is the process by which data in a dataset are transformed, or changed, during data cleaning and involves the use of mathematical operations to reveal features of the data that are not observable in the data's original form [57]. The usual process of data transformation involves converting data structures, files or database contents, and documents. Knowledge transformation is often considered either a part of this broader concept of data transformation, or a special case of it, because ultimately semantic knowledge representation also boils down to managing syntactic data [58]. The focus of knowledge transformation is mainly on methods and techniques that allow the extraction of knowledge from data.

Often, data conversion also involves software conversion from one computer language to another, to make the running of a particular software tool possible on a different platform [59]. This is often referred to as data migration or software migration. However, as reflected in the literature, data transformation means data processing for some authors, whereas it is synonymous with data translation or data integration for others. We adopt the comprehensive interpretation according to which data transformation blends data preprocessing, data processing (transformation), and data post-processing (presentation). Data transformation involves multiple steps, from the aggregation of data, through cleaning, classification, and interpretation of data, to extraction of patterns to be evaluated and representation of data patterns. These steps are needed to support and interpret changes in the structure, representation, and content of data [56].

Since the beginning of the big data era, many authors have addressed the challenges that they have faced. The primary issue is not the huge amount and diversity of data, but how to transform and extract valuable insights from dumped and dynamically changing data [41]. The aim is to discover previously unknown interrelations among unrelated attributes of datasets [60] [61]. Many books have been published on the essence and challenges of data transformation processes [5]. The complexity and the other challenges of data mining, as well as the steps needed to find solutions, have been addressed from many aspects by data scientists and analysts [62] - [64]. The difficulty and specific challenges of correlation analysis (i.e. finding the measures, the degree of association of the data, or the strength of the relationship among them) using mathematical operations are also addressed in the literature [65]. These challenges include (i) spatial and temporal variation of data, (ii) missing values, and (iii) the lack of balance in sampling [66]. Finding answers to these challenges is a concern not only for researchers but also for business and market managers and decision makers who should base their decisions and actions on the insights gained from big data [67]. To this end, they have to understand the aggregation of big data, the approaches to extracting patterns, and the use of them to predict future situations and/or behaviors.

The ultimate objective of big data management is to generate new business opportunities for service industries. However, since more than 80% of the world's data is unstructured, most businesses do not even try to use it for their benefit [68]. Big data itself has no real meaning unless it is exploited by having information and knowledge extracted from it. Various authors argue that not only the amount of data is important but also their diversity, which may reveal various semantic patterns [69]. There is a wide variety of big data sources. They range from natural processes and substances through engineered physical processes and artifacts to virtual environments and objects. A recent source of big data has been social media and websites [70]. Data collected from diverse sources and represented in various formats need to be transformed (prepared for semantic processing). This constitutes an essential step in the process of a meaningful analysis [71].

Social media and website data need preprocessing that involves common steps such as aggregation, cleaning, and sorting, because real-world data are typically impure [72]. Preprocessing is also expected to (i) determine the accuracy and completeness of data [73], (ii) reduce noise in data and correct the omissions, and (iii) handle missing values [74]. In addition, preprocessing includes activities such as (i) classification, which increases the efficiency of data retrieval [75]; (ii) clustering, which facilitates structured handling of data [76] [77]; and (iii) visualization, which can be applied to knowledge discovery processes [78] as well as to the results of other transformative actions [79]. Data cleaning (~ data checking or ~ validation) is regarded as an important process by which missing, erroneous, or invalid data are determined and cleaned, or removed, from a dataset, and it follows the data preparation process. Although the concept of back transformation (the process in which mathematical operations are applied to an already transformed dataset to revert the data to their original form) is known in the literature, the number of papers on the topic is much lower than the number dealing with forward transformations [80].

An important action in the transformation of data to knowledge is data mining, which has gained increasing attention since the beginning of the big data era [4]. Data mining is defined as "an algorithmic process that takes data as input and yields patterns such as classification rules, association rules, or summaries as output" [81]. It seems to be a simple action, but its implementation is challenging due to the associated computational complexity [82]. Data mining tasks may be used to discover the knowledge and rules hidden within MoLD, such as product usage patterns, maintenance history, customer support information, updated bills of material, and updated product demand information [83].

Many researchers argue that the challenge of data mining is partially caused by the need for effective algorithm designs that can tackle mining problems even for huge volumes of complex and dynamic data [84]. This huge dimensionality of data, together with the explosion of features and variables, is what brings new challenges to data analytics [85]. Various methods have been developed to extract meaningful knowledge from complex and dynamically changing data [86] [87]. Regarding MoLD processing, the literature diverges from the methods of data processing. For example, failure modes and effects analysis (FMEA) has been used for identifying design problems during the MoL phase. FMEA-based methods have a critical weak point in that they do not consider product degradation in failure models quantitatively [88].

As discussed in [84], "while typical data mining algorithms require all data to be loaded into the main memory, this is becoming a clear technical barrier for big data because moving data across different locations is expensive, … even if we do have a super large main memory to hold all data for computing." To solve the complexity problem of data mining, some authors have proposed applying parallel computing [89] [90]. Others prefer collective mining of sample and aggregate multisource data and then using parallel computing in the actual mining process [91]. Preprocessing plays an important role in the case of big data [92], but it is a time consuming set of activities, and there are certain threads associated with it [93]. The main transforming activity is data analytics, which may have a range of objectives, such as obtaining useful values, extracting patterns, providing suggestions, and optimizing decision-making [51] [52]. Real-time processing of big data remains a very challenging task [84]. According to McKinsey [94], smart data analytics will be the key to competition, productivity, and innovation.

2.3.2. Lessons learned

Thanks to the development of automatic identification, data capture, and storage technologies, people generate data much more quickly and collect much more data than ever before in business, science, engineering, education, and other areas [95]. Current literature identifies the main activities in processing complex and unstructured data as (i) data preparation, (ii) data mining, (iii) pattern evaluation, and (iv) knowledge representation [96]. The change in the nature and the sources of data implies a change in the steps of data transformation [97]. Given the large dynamics, data processing should consider the time-dependent validity of data [98]. The literature claims that there is a need for data transformation techniques able to manage changing data patterns or to perform dynamic pattern recognition and evaluation [32]. Many papers explained that big data is mainly unstructured and heterogeneous, but its cleaning and preprocessing may lead to a relatively high loss of data [56]. It is also claimed that a significant amount of big data is discarded [98], which may greatly influence the results of data transformation, but no clear solution has yet been proposed concerning these issues, which points to the complexity challenges of big data.

The mining of big data primarily focuses on extracting patterns to be evaluated by both manual and automated approaches [99]. Currently, dealing with patterns requires multiple expert interventions (especially after mining) [100]. At the same time, the currently widespread methods for big data transformation do not consider human behavior, which adds uncertainty to the outcome of the process [101]. Extraction of patterns is typically done based on historical data rather than based on real-time acquired data. Mining and transforming big data necessitates highly scalable strategies [102]. To achieve more effective processing, the literature suggests developing sophisticated data filtering and integration techniques, as well as using advanced parallel computing environments and more effectively involving users [4].

Researchers also observed that if the process of transforming data to knowledge is time consuming, this delay may reduce the relevance of the extracted knowledge and its validity in the dynamic context, or it can even make the extracted knowledge invalid [103]. This issue has been addressed by many publications, but the contour of a general solution does not seem to be emerging. However, one issue that does not seem to be sufficiently addressed in the literature is extracting meaning from data (automatically or semi-automatically). The issue is important because it concerns and may computationally influence all data transformation steps. A hierarchy of concepts interlinked by the assumed relationships, along with axioms that can express the relationships of the concepts and constrain their interpretation, are seen as ingredients of a possible solution [104]. In addition, only limited efforts have been made to capture the semantics of transformed data and to give transformed data meaning in context [105]. Insufficient attention has been given (i) to the relationships between signifiers such as words, phrases, signs, and symbols; (ii) to what they stand for; and (iii) to what their denotations are [106].

It seems that there are multiple challenges related to the early preparatory activities of data analytics. One of them is data inundation, which may manifest as the major performance bottleneck for processing (cleaning, sorting, structuring, etc.) the output of increasingly complex sensor networks used to monitor product use and lifecycle performance [107]. In addition to the amount of data generated by sensors, the signals and data generated by the end-user products themselves should also be accounted for. It was argued that data analytics will be significantly challenged by the need to combine sensor-generated (objective and aggregated) macro data with end-user-generated (subjective and finely granular) micro data to determine their mutual meaning and impacts [108] [109]. Furthermore, associating quantitative (measured and factual) data with qualitative (provided by social networks) data is needed. Data modeling approaches are often distinguished as exploratory and confirmatory approaches [110], [111]. Finally, distinction is made between nonparametric statistical confirmatory data modeling and parametric statistical confirmatory data modeling [112].

2.4. Knowledge aggregation about the means for data transformation

2.4.1. Mapping of data transformation tools and packages

The literature presents, discusses, and compares many tools that have been developed to help us understand and process data. The overwhelming majority of these tools are general-purpose statistical tools [113]. A smaller number of tools have been developed to assist in improving products and services [114] [115]. The general-purpose software means are typically sorted into three categories: (i) single-task-oriented software tools [116], (ii) multitask-oriented integrated software packages (and toolboxes) [117], and (iii) multifunctional development environments [118]. The first category consists of software implementation of algorithms, procedures, or techniques to represent, enhance, analyze, or transform input and output data. A second-level categorization of the single-task-oriented software tools is made based on the types of tasks they are intended to support. Typical representatives are business analysis tools, data visualization tools, and trend analysis tools, which are marketed by many vendors.

The second category of systems includes (often-modularized) software packages that combine and interlink functionalities of multiple software tools. The component tools typically share a common user interface and can exchange data with each other [119]. The statistical procedures offered by the integral packages for exploring and predicting from big data address the issues of big data processing (BDP) such as source heterogeneity, noise accumulation, spurious correlations, and incidental endogeneity, in addition to balancing statistical accuracy and computational efficiency [120]. The third category includes developer tools that are able to generate partly or fully automated algorithms, source codes, and executable programs, and to interlink these. The term "environment" in the name of the category indicates that all developer tools and production servers are incorporated.

There is a need for computational theories and tools to assist humans in servicing [121], and in extracting useful information and knowledge from, the rapidly growing volumes of digital data [122]. This is confirmed by a 2017 study explaining that the large volume of the data makes it difficult for human beings to extract valuable knowledge from it without powerful tools [123]. Notwithstanding that the literature discusses many big data mining and analysis tools, most are still in their infancy [124]. The traditional tools are able to capture, curate, analyze, and visualize big data, but usually fail to fulfill the full variety of needs such as (i) satisfying all experimental designs [125], (ii) finishing the processing in a reasonable time [126], (iii) functioning in parallel [127], and (iv) being scaled to large datasets [128]. The fact is that as the number of existing commercial (Cl) and open-source (OS) tools grows, choosing the most appropriate one for a particular data analysis task is becoming increasingly difficult [129]. An apparent technological issue for traditional software tools is that the amount of data generated and stored in different sources grows rapidly, and their handling needs a sufficient level of automation [130]. Lacking this, it is becoming harder to capture, store, manage, analyze, visualize, and share mass data using typical tools [131]. Required are powerful and efficient tools that extract useful information from data and that can cope with big data challenges [132].

It has been found that certain software tools solely operate as information providers to data mining tools and do not support any analysis functions [54]. Others, as argued, "can be abused for data mining, but their intended use lies somewhere else" (such as software packages dedicated to pure statistical analysis or Matlab's Neural Network Toolbox) [133]. Other software tools are advertised as data mining or knowledge discovery tools, but they only do reporting and visualization, such as Oracle Discoverer [134]. The literature includes surveys and comparative studies that analyze the functional capabilities of data analytics tools and compare the performance of these tools [135] [136]. A survey done in 2016 reported that the top ten tools were as follows: R, Python, Microsoft SQL Server, Microsoft Excel, RapidMiner, Hadoop, Spark, Tableau, KNIME, and Scikit-learn [137]. In Table 2.1, we sort the tools we investigated in our study according to the functions they offer for data analytics. Clearly, any comparison of the tools must consider not only the functionalities and the data analytics tasks at hand but also the user groups, the data structures, the processing methods, the import and export of data, and the use of models as well as the platforms and the licensing.

Other web resources showed that several pragmatic issues, such as the budget and the user experience, also influence the choice of tools.[1] As an overall finding, we can argue that there is no single tool, not even an integrated package, that could cover all needs and steps of data analytics, in particular not in the case of BDP and applications [138]. It seems to be a generally accepted conclusion in the literature that no tool is better than the others are [139], and that users can select the adequate data analytics software package only based on a critical analysis of the objectives and the application case [140]. It is worth noting that there are also several quasi-data analytics tools, which we have not considered relevant for our specialized study. In the area of engineering, for example, are tools such as ThingWorx[2] (PTC), Exaled[3] (Dassault Systemes), and Omneo[4] (Siemens).

2.4.2. Lessons learned

Much work has been done to develop and enhance data analytics tools. The literature emphasizes that big data cannot be managed with traditional methodologies or data mining software tools [165] [166]. In general, they have great difficulty handling heterogeneity, volume, and speed, as well as privacy and accuracy, and they are inadequate for addressing such characteristics [167]. Applying existing data mining algorithms and techniques to real-world problems raises many challenges due to the inadequate scalability and other limitations of these algorithms and techniques [168] [169]. Several authors confirm that there is a need for new computational theories and tools to assist humans in extracting useful information or knowledge from the rapidly growing volumes of digital data [170].

There are a multitude of tools that can help in understanding and interpreting various application data. They can also help improve products and services based on dedicated data transformation steps. It is often mentioned that even the most sophisticated tools need human interaction [171]. Another open issue is that it is not clear how a data analytics system can deal simultaneously with both historical data and real-time data, nor is it clear what the optimal architecture of such a system would be [172]. A less significant but still important issue is that it is difficult to find user-friendly visualizations for cases involving large data volumes [173].

[1] https://www.softwareadvice.com/bi/data-analysis-comparison/
[2] https://www.ptc.com/en/resources/iot/product-brief/thingworx-platform
[3] https://www.3ds.com/products-services/exalead/
[4] https://community.plm.automation.siemens.com/t5/Digital-Transformations/Use-Omneo-Big-Data-analytics-to-gain-product-performance/ba-p/359386

Table 2.1. Investigated data analytics software tools

Software tool	*License*	*Data transformation steps allowed by the software*
ADaMSoft [141]	OS	Data classification, data mining, data visualization
Analytica [142]	Cl	Data visualization, simulation
BV4.1 [143]	OS	Pattern detection, data visualization
CLUTO [144]	OS	Data clustering
COMSOL [145]	Cl	Data modeling, simulation, data visualization
Dataiku [146]	OS	Data visualization, data preprocessing, data modeling
DataMelt [147]	OS	Data visualization, data preprocessing, data mining
FreeMat [148]	OS	Data processing, data visualization
GNU Octave [149]	OS	Data preprocessing, data visualization
JASP [150]	OS	Data processing, pattern recognition, data visualization
KNIME [151]	OS	Data cleaning, data classification, data visualization
MATLAB [152]	Cl	Data preprocessing, data mining, pattern evaluation, data visualization
MaxStat [153]	Cl	Statistical analysis of data
Microsoft Excel [154]	Cl	Data preprocessing, data mining, pattern evaluation, data visualization
OpenStat [155]	OS	Simulation
Oracle [156]	Cl	Data preparation, data classification, data clustering, data mining
R [157]	OS	Data preprocessing, data mining, pattern evaluation, data visualization
RapidMiner [158]	Cl	Data preparation, data mining, data visualization
SAS [159]	Cl	Statistical data analysis
Scilab [160]	OS	Statistical data analysis
Shogun [161]	OS	Data preprocessing, data mining, pattern evaluation, data visualization
SPSS Statistics [162]	Cl	Data preprocessing, data mining, pattern evaluation, data visualization (mainly statistical analysis)
Stata [163]	Cl	Data visualization, pattern evaluation
WEKA [164]	OS	Data classification, data clustering, data mining, attribute selection, data cleaning, data visualization

2.5. Knowledge aggregation about applications of data analytics

2.5.1. Mapping of data analytics applications

Data-driven applications have emerged in the last decades [174]. Big data is rapidly expanding in all science and engineering domains, as well as in physical, biological, and biomedical sciences [84]. Research has provided important information for designing big data mining algorithms and systems [175]. Because every discipline and application domain has a stake, big data has become primordial for multidisciplinary problem solving [176]. In this case, the challenge is how it is possible to use data regardless of

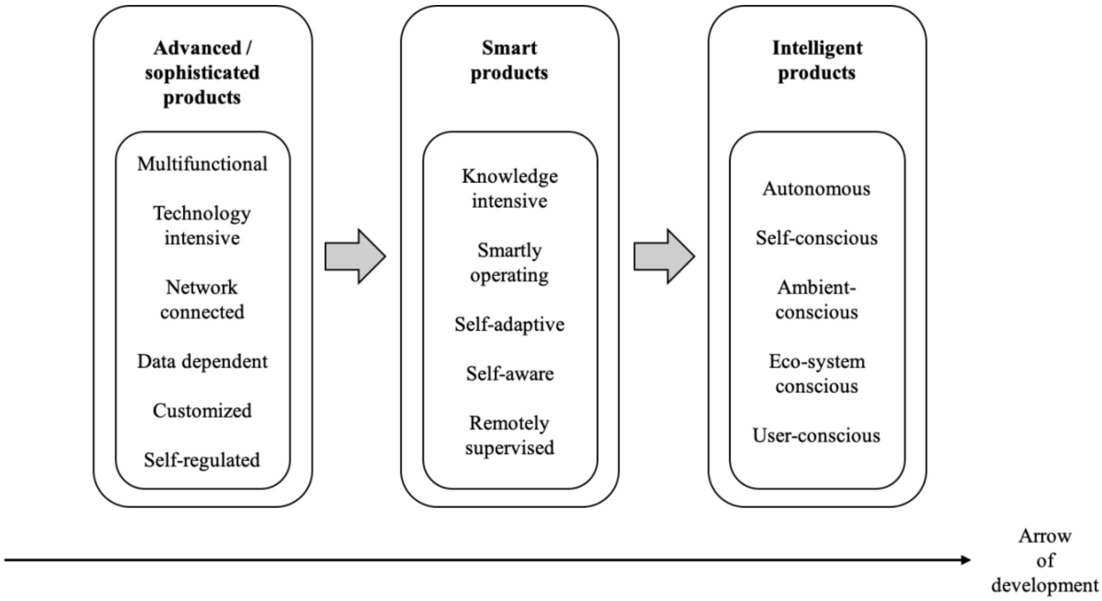


Figure 2.5. The dominant trend of development in new generations of products

the application domain [177]. Despite the complexity, the serious efforts scientists and developers are currently making to explore what big data can offer have created an optimistic atmosphere. “Many big data applications will have unintended and unpredictable results as the data scientist seeks to reveal new trends and patterns that were previously hidden” [178].

In spite of the above efforts, the domains of engineering- and product-associated BDP are behind the overall progress because of the sheer fact of late recognition. Another issue is the rapid paradigmatic changes in the field driven by the converging technologies and the embedded software and cyber-ware in practically all products. Currently already observable, these imply many changes [179]. First, engineered products are becoming more multifunctional, technology intensive, network connected, data dependent, customized and personalized [180]. Operation and maintenance process data can be tracked in real time during a product’s MoL stage by embedding an information device into the product itself [83]. Products with these characteristics are often referred to as advanced or sophisticated products. However, the largest paradigmatic change is that they are rapidly becoming knowledge-intensive and operating smartly, or even progressing towards some forms of intelligent operation (Figure 2.5) [181]. Therefore, in line with many other researchers, we considered advanced durable consumer products as a specific application domain of big data and data analytics [182].

Interestingly, there seems to exist a debate as to whether BDP is relevant to all application domains, questioning whether BDP has equal importance in the various data-intensive application domains [177]. Some voices claim that BDP exists only on paper, as a theoretical perception that cannot be put into practical applications. Others argue that BDP is still in a rather premature state, as it still struggles with complex application challenges and cannot provide immediate benefits for practical applications [183]. The premature state is associated with the lack of both sophistication and dependability of the implementation technologies and the low level of elaboration of application methodologies relevant for various application domains. On the other hand, there are already numerous examples demonstrating benefits and advantageous changes brought by BDP to both professional application domains and daily life. In the context of product development, BDP has opened opportunities not only for storing data about customers but also for analyzing large volumes of data about their behaviors and customs, which in turn can be used to gain competitive advantage [184].

Current typical big data applications are mainly related to processing sensitive information and data exchanges or transmissions [185], [186]. In electronic commerce, big data analysis was concluded to be elemental for the success of websites, since it facilitates building markets and increasing customers' abilities to extract relevant information on the web [187]. In financial trading, BDP permits service companies (such as Google) to gain profit by making use of data [188]. BDP has also proved beneficial in providing a multipurpose data processing system to support financial transactions and services [189]. In social administration and government, extracting informative data and knowledge patterns provides opportunities to improve productivity and to increase the level of effectiveness [94] as well as to forecast in advance and take actions in case of natural damages [190]. In the field of health care, online diagnosis repositories are one successful application of big data [191].

BDP also helps decrease variability in healthcare quality and augments healthcare's effectiveness with data mining techniques (for instance, to determine the most effective treatments for different conditions) [192]. In the pharmaceutical industry, collecting big data can provide information about, for example, preferences for certain medicines and drugs [193]. In telecommunication, big data mining has been applied to illuminate use trends and habits and to identify telecommunication fraud [194]. Likewise, preprocessing big data has been confirmed to boost the plausibility and accuracy of forecasts [195]. In scientific research, many fields have become highly data-driven due to the development of computer science [196], such as astronomy [197], social computing [198], bioinformatics [199], and biology [200], which generate enormous datasets able to provide the basis for inquiry or to drive the whole system design when analyzed [201]. In all of the mentioned applications, and in others, significant challenges are related to system capabilities, algorithmic designs, and design models [202].

2.5.2. Lessons learned

As argued in [84], "Driven by real-world applications … managing and mining big data have been shown to be a challenging yet very complicated task." Practically independent of the field of application, one of the main challenges of BDP is exploring large volumes of data and extracting useful information or knowledge to guide future actions [203]. Notwithstanding, new applications are revealed and new approaches are proposed. One proliferating field of application is using BDP in strategic product development and lifecycle engineering. In this particular field of application, rapid changes are predominant. We can witness the current trend of the intellectualization of products and services [204]. Intellectualization refers to equipping industrial and consumer products with digital connectivity and data communication functions as well as with capabilities that mimic human-type intelligence. The first developments have been supported by IoT technologies as overall infrastructure. However, it has also been clarified that the IoT enables connectivity and information exchange rather than implementation of operational intelligence of products [205]. It is the role of cyber-physical systems and the various advanced forms of artificial (system) intelligence to provide sophisticated mechanisms for building situation and context awareness, for reasoning and decision-making, and for adaptation to operational situations and objectives [206].

The term "intelligence" has become popular in both the scientific and the professional literatures, though it begs for more careful usage in the context of artifacts and services. As for now, terms such as "advanced," "sophisticated," "smart," "autonomous," and "intelligent" are used interchangeably, as well as indistinguishably, by various authors [207]. Until now, the concept of intelligent products has remained fuzzy, and the use of the term is confusing [208]. Interestingly, even the scientific literature is divided in terms of using these terms to characterize the operation and/or behavior of artifacts and their interaction with humans and other artifacts (systems) [209]. There seems to be a problem with the verbatim interpretation of the term "intelligent" as well as with the relationships of intelligent products to knowledge acquisition and processing. This entails the need for further work the considers the variety of application contexts. In addition, there is a need for a new classification of these products that simultaneously considers the achieved level of intelligence and the specific manifestations of these levels [210]. On the other hand, researchers active in various fields of intelligent products do agree that there is still a long way to go before different kinds of machines and systems will be able to intelligently communicate, reason, and understand each other [211] [212]. Some of these researchers believe producing truly "intelligent" tools will require more than what is typically provided by ontologies and semantic web-related technologies [213] – [215].

2.6. Discussion of the findings and conclusions

2.6.1. Synthesis of the findings

From a philosophical perspective, the whole of our inquiry was driven by pragmatism, a doctrine that entails setting a concrete goal and acting purposefully towards achieving it. Pragmatism also meant that, rather than reviewing all pertinent publications, we considered only those that were highly relevant and significant from the perspective of our ultimate research objective. Furthermore, the observed trends and the proposed theories and solutions were mainly evaluated in terms of the caused changes and their success in practical applications. This approach lent itself to a reflexive review, which is appended by a concise discussion of our prospective future research.

Based on a statistical and relational study of the literature, we derived a reasoning model that identified and brought four domains of knowledge into an implicative interrelationship. The four domains are (i) changes in the nature of data and their characteristics; (ii) approaches of data analytics-based transformations; (iii) data analytics algorithms, tools, and packages; and (iv) representative application fields and practices of data analytics. A specific objective of our study was to synthesize knowledge for a fifth domain of interest, which is contributions to data analytics-based support of product enhancement and new product innovation. Actually, this objective created an application context for the whole of the explorative study. The findings (i.e.

the specific pieces and chunks of knowledge obtained from various sources) were synthesized in this context. By bringing the five domains of knowledge into implicative relationships, the adopted reasoning model entailed a kind of natural streaming of knowledge that in turn allowed us to build an intellectual platform for a new, sufficiently tailored supporting means.

Current data analytics should deal with data that are largely different from those processed digitally some decades ago. The major difference is not only the amount of data but also the complexity of data. Although this creates new challenges for data analytics, it also creates opportunities for new value creation approaches. There seems to be a consensus in the literature that the overwhelming majority of existing (traditional) data processing methodologies and tools cannot properly address the complexity of big data, and that exploiting the affordances of big data in various application contexts needs a stronger contextualization of data transformation processes. The transformation techniques and tools are expected to support real-time processing of data and the highest possible level of semantic interpretation of data. Time-dependent (and real-time) processing of complex data streams still raises many issues, in addition to the well-known issues of storing big data, fusion of heterogeneous multi-data sources, and visualization of big data.

The use of cloud computing methods and resources in capturing and processing big data is becoming a daily standard, and it is exploited in many areas of big data. This phenomenon rapidly proliferates in these days since users are able to access data and data-processing tools from a cloud anywhere and anytime they are needed [216]. Actually, several existing data processing applications need cloud environments, such as distributed multimedia data management. In turn, the need for efficient BDP also raises many new requirements for cloud computing, for instance (i) resources for handling large-scale heterogeneity, (ii) methods for effective and smart multimedia content retrieval, (iii) transport and security protocols, and so on. Large databases with large volumes of vaguely related data entities or complex data structures are also the focus of research. The intention is to lessen data uncertainty and to increase understanding of meaning and consequently to enhance the reliability of analysis and decision-making. Since the amount of data grows irresistibly, data subsampling is becoming a means of resolving computational limitations. Although surrogating entire complex datasets helps overcome the real-time constraint, it introduces even greater uncertainty.

It can be predicted that the efficiency and reliability of data mining and knowledge discovery will remain the major issues for advanced big data analytics. Processing algorithms and mechanisms should be based on new underpinning theories that allow us to manage the volume, the distribution, the cognitive complexity, and the dynamically changing characteristics of big data. The time characteristic of big data does not seem to a significant obstacle, but the interplay among all aspects of big data does. Across all industries, big data is a new business asset, and advanced data analytics will help businesses to become smarter, more productive, and better at making predictions. In the context of future product and service development, new sources of data such as social media will offer new opportunities for designers to gain insights into consumers' purchasing preferences, decisions, and behavior and new opportunities to uncover information in context in ways that are not possible with traditional product functionalities and lifecycle data management approaches. In a wide field-based collecting data, designers may rely on IoT principles and technologies, while in terms of locally interpreting and reasoning with big data, they may rely on smart cyber-physical systems technologies.

To resolve the incessant problems of big data management and processing, some authors have proposed specific solutions for a selection of tools for coping with the complexity of big data in particular application domains. A purposeful regrouping of these solutions (touched upon in previous sections) could reveal the fact that most authors are committed (if not attached) to real-time analysis of data and to developing powerful tools and better system architectures so that companies making durable consumer products can realize value by understanding their operations, customers, and distributors and the marketplace as a whole.

2.6.2. Conclusions

We formulate our main conclusions as follows:

- The completed literature study reinforced our observation that all aspects of the current daily practice of data analytics are changing and developing rapidly. The source phenomenon triggering these changes in the methods, tools, and applications is the change in the nature of digital data. This change manifests itself in the growing amount and increasing complexity of data, which challenge pattern-based information and knowledge mining.
- Significant diversification can be observed in the area of data transformation. A plethora of methods and techniques have been developed for systematic and controlled data aggregation, interactive visualization, structural and semantic interpretation, and trend analysis and prediction. However, many of these are general (mathematical and statistical) approaches that do not reflect the specific needs of particular applications.
- Diversification of methods and techniques is naturally followed by the diversification and articulation of DATs and systems. Articulation is reflected in the fact that the commercialized enablers range from (i) specific-purpose (individual) tools, through (ii) multipurpose (integrated) packages, to (iii) application-oriented toolboxes. The literature reports several interoperability and efficiency issues.
- Design application of data analytics seems to be in a premature phase. As a combined effect of the proliferation of DATs and the IoT connectivity, companies are gradually recognizing the opportunities and trying to convert them into business benefits. However, neither comprehensive methodologies nor dedicated toolboxes seem to be available to facilitate their endeavors.
- There is a kind of paradoxical situation in the large number of data analytics tools and their under-exploitation in product development and innovation. In other words, product managers, designers, and developers need to be supported by proper data analytics enablers in order to extract new knowledge and achieve significant benefits by processing MoL product data [217]. It seems to be a pragmatic but instrumental

strategy to combine the existing tools and packages into user-friendly and application-sensitive toolboxes.

- There is a need to investigate data analytics in the context of processing MoLD, which are typically characterized by huge variety and dynamics and a multiplicity of relationships. Our study revealed that only a very limited number of papers specifically address issues of extracting knowledge from this kind of data, despite the above characteristics and the purpose of providing information about product usage and operations. The existing literature lacks a systematic and extensive analysis of stakeholders within the MoL phase [218]. Moreover, analyzing MoLD is a challenge and is still in its infancy, since collection of MoLD reveals several issues [219]. In terms of available tools, we find not one specifically developed for processing MoLD.
- To improve products and services, designers need to consider the application context and objective in transforming raw big data into creative knowledge. This transformation should also help them make proper decisions as to the best enhancement opportunities. However, no tools currently available on the market are dedicated to the changing components of design tasks. In addition, the use of some of the tools is complicated because they require the user to have a certain level of knowledge and skills to write and employ algorithms for data analysis. Another critical issue from the perspective of designers is the lack of data integration and the lack of abstraction to semantic interpretation of the outputs, without which it is difficult to put the results into a specific design context.
- A smart data processing system to process real-time data streams to support the operator is becoming a necessary tool to handle the vast amount of data generated by online instruments. It will provide the benefit of prefiltering useful data to be transferred from the remote monitoring system and stored for reference purposes. Many of these data are stored but never accessed, and the potential of expensive instruments often goes unrealized due to a lack of understanding and to difficulty in extracting useful information from massive databases [220].

2.7. References

[1]. Shu, H. (2016). Big data analytics: six techniques. *Geo-spatial Information Science*, 1-10.

[2]. Davenport, T.H., & Prusak, L. (1998). Working knowledge: How organizations manage what they know. *Harvard Business Press*, Buston, MA.

[3]. Bufardi, A., Kiritsis, D., & Xirouchakis, P. (2008). Generation of design knowledge from product life cycle data. *Methods and Tools for Effective Knowledge Life-Cycle-Management*, Springer, Berlin, 375-389.

[4]. Che, D., Safran, M., & Peng, Z. (2013). From big data to big data mining: challenges, issues, and opportunities. In: *Proceedings of the International Conference on Database Systems for Advanced Applications*, Berlin Heidelberg, 1-15.

[5]. Mayer-Schönberger, V., & Cukier, K. (2013). Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt.

[6]. Xhafa, F., & Barolli, L. (2014). Semantics, intelligent processing and services for big data. *Future Generation Computer Systems*, 37, 201-202.

[7]. Kiritsis, D. (2004). Ubiquitous product lifecycle management using product embedded information devices. In: *Proceedings of the International Conference in Intelligent Maintenance Systems*, x-x.

[8]. Guttag, J. (1977). Abstract data types and the development of data structures. *Communications of the ACM*, 20(6), 396-404.

[9]. Brodie, M.L. (1984). On the development of data models. *On conceptual modelling*, Springer, New York, 19-47.

[10]. Ballard, C., Herreman, D., Schau, D., Bell, R., Kim, E., & Valencic, A. (1998). Data modeling techniques for data warehousing. *IBM Corporation International Technical Support Organization*, 25.

[11]. Gunn, S.R., Brown, M., & Bossley, K.M. (1997). Network performance assessment for neurofuzzy data modeling. *Lecture Notes in Computer Science*, 1280, 313-323.

[12]. Jia, N., Xie, M., & Chai, X. (2011). Development and implementation of a GIS-based safety monitoring system for hydropower station construction. *Journal of Computing in Civil Engineering*, 26(1), 44-53.

[13]. Jukic, N. (2006). Modeling strategies and alternatives for data warehousing projects. *Communications of the ACM*, 49(4), 83-88.

[14]. Worboys, M.F., Hearnshaw, H.M., & Maguire, D.J. (1990). Object-oriented data modelling for spatial databases. *International Journal of Geographical Information System*, 4(4), 369-383.

[15]. Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modelling versus ontology engineering. *ACM SIGMod Record*, 31(4), 12-17.

[16]. Meersman, R. (1999). The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In: *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications*, 1-14.

[17]. Ramakrishnan, R., & Gehrke, J. (2000). Database management systems. McGraw Hill.

[18]. Siau, K., Nah, F.F.-H., & Cao, Q. (2013). A meta-analysis comparing relational and semantic. *Innovations in Database Design, Web Applications, and Information Systems Management*, 394-409.

[19]. Verhoef, P.C., Spring, P.N., Hoekstra, J.C., & Leeflang, P.S. (2003). The commercial use of segmentation and predictive modeling techniques for database marketing in the Netherlands. *Decision Support Systems*, 34(4), 471-481.

[20]. Goodchild, M.F. (1996). The application of advanced information technology in assessing environmental impacts. *SSSA Special Publication*, 48, 1-18.

[21]. Ter Bekke, J.H. (1992). Semantic data modeling. Prentice Hall, Hemel Hempstead.

[22]. Ter Bekke, J.H. (1995). Meta modeling for end user computing. In: *Proceedings of the DEXA Workshop*, 267-273.

[23]. Ter Bekke, J.H. (1997). Comparative study of four data modeling approaches. In: *Proceedings of the 2nd International EMMSAD Workshop*, Barcelona, B1-B12.

[24]. Molenaar, M. (1996). A syntactic approach for handling the semantics of fuzzy spatial objects. *Geographic Objects with Indeterminate Boundaries*, 2, 207-224.

[25]. Shoval, P., & Shiran, S. (1997). Entity-relationship and object-oriented data modeling - An experimental comparison of design quality. *Data & Knowledge Engineering*, 21(3), 297-315.

[26]. Coronel, C., & Morris, S. (2016). Database systems: Design, implementation, & management. *Cengage Learning*, Boston, MA.

[27]. Post, G.V. (1999). Database management systems: designing and building business applications. McGraw-Hill, Boston, MA.

[28]. Andreeva, P. (2006). Data modelling and specific rule generation via data mining techniques. In: *Proceedings of the International Conference on Computer Systems and Technologies*, IIIA.17-1-IIIA.17-6.

[29]. Henderson, D.J., Carroll, R.J., & Li, Q. (2008). Nonparametric estimation and testing of fixed effects panel data models. *Journal of Econometrics*, 144(1), 257-275.

[30]. Madden, S. (2012). From databases to big data. *IEEE Internet Computing*, 16(3), 4-6.

[31]. Dietrich, D. (2015). Data science & big data analytics: discovering, analyzing, visualizing and presenting data. Wiley.

[32]. Kriegel, H.-P., Borgwardt, K.M., Kröger, P., Pryakhin, A., Schubert, M., & Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, 15(1), 87-97.

[33]. Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META Group Research Note*, 6, 70.

[34]. Russom, P. (2011). Big data analytics. *TDWI Best Practices Report*, Fourth Quarter, 1-35.

[35]. Fouad, M.M., Oweis, N.E., Gaber, T., Ahmed, M., & Snasel, V. (2015). Data mining and fusion techniques for WSNs as a source of the big data. *Procedia Computer Science*, 65, 778-786.

[36]. Rahman, H., Begum, S., & Ahmed, M.U. (2016). Ins and outs of big data: A review. In: *Proceedings of the International Conference on IoT Technologies for HealthCare*, 44-51.

[37]. Jara, A.J., Genoud, D., & Bocchi, Y. (2014). Big data for cyber physical systems: An analysis of challenges, solutions and opportunities. In: *Proceedings of the 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 376-380.

[38]. Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A., & Buyya, R. (2015). Big data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 79, 3-15.

[39]. Yu, P.S. (2013). On mining big data. *Web-Age Information Management*, 7923.

[40]. Jun, H.-B., Kiritsis, D., & Xirouchakis, P. (2007). Research issues on closed-loop PLM. *Computers in Industry*, 58(8), 855-868.

[41]. Cassina, J. (2008). Extended product lifecycle management. Doctoral dissertation, Politecnico di Milano.

[42]. Madhikermi, M., Buda, A., Dave, B., & Främling, K. (2017). Key data quality pitfalls for condition based maintenance. In: *Proceedings of the 2nd International Conference on System Reliability and Safety*, IEEE, 474-480.

[43]. Vladimirova, D., Evans, S., Martinez, V., & Kingston, J. (2011). Elements of change in the transformation towards product service systems. *Functional Thinking for Value Creation*, 21-26.

[44]. Ericson, A., Müller, P., Larsson, T., & Stark, R. (2009). Product-service systems–from customer needs to requirements in early development phases. In: *Proceedings of the 1st Industrial Product Service Systems Conference*, 62-67.

[45]. Shin, J.H., Jun, H.B., Catteneo, C., Kiritsis, D., & Xirouchakis, P. (2015). Degradation mode and criticality analysis based on product usage data. *The International Journal of Advanced Manufacturing Technology*, 78(9-12), 1727-1742.

[46]. Ward, J.S., & Barker, A. (2013). Undefined by data: a survey of big data definitions. Retrieved from https://arxiv.org/abs/1309.5821.

[47]. Bollier, D., & Firestone, C.M. (2010). The promise and peril of big data. *Aspen Institute, Communications and Society Program*, Washington, DC., 56.

[48]. Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the Future*, 2007(2012), 1-16.

[49]. Shah, M. (2016). Big data and the internet of things. *Big Data Analysis: New Algorithms for a New Society*, Springer, 207-237.

[50]. Mohammed, E.A., Far, B.H., & Naugler, C. (2014). Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends. *BioData Mining*, 7(1), 22.

[51]. Nguyen, S.H., Skowron, A., & Synak, P. (1998). Discovery of data patterns with applications to decomposition and classification problems. *Rough Sets in Knowledge Discovery 2*. Springer, 55-97.

[52]. Silverston, L., & Agnew, P. (2009). Universal patterns for data modeling. The Data Model Resource Book, John Wiley & Sons.

[53]. West, M. (2011). Developing high quality data models. Elsevier.

[54]. Goebel, M., & Gruenwald, L. (1999). A survey of data mining and knowledge discovery software tools. *ACM SigKDD Explorations Newsletter*, 1(1), 20-33.

[55]. Lee, S.-J., & Huh, E.-N. (2013). Shear-based spatial transformation to protect proximity attack in outsourced database. In: *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 1633-1638.

[56]. Rahm, E., & Do, H.H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4), 3-13.

[57]. Berti-Equille, L., Dasu, T., & Srivastava, D. (2011). Discovery of complex glitch patterns: A novel approach to quantitative data cleaning. In: *Proceedings of 27th International Conference on the Data Engineering*, IEEE, 733-744.

[58]. Czarnecki, K., & Helsen, S. (2003). Classification of model transformation approaches. In: *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, Anaheim, x-x.

[59]. Sharma, S., & Srinivasan, S. (2014). Issues related with software conversion projects. *International Journal of Electrical Electronics & Computer Science Engineering*, 1(4), 10-13.

[60]. King, E.A. (2005). How to buy data mining: A framework for avoiding costly project pitfalls in predictive analytics. *Information Management*, 15(10), 38.

[61]. Witten, I.H., Frank, E., Hall, M.A., & Pal, C.J. (2016). Data mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[62]. Berson, A., & Smith, S.J. (1997). Data warehousing, data mining, and OLAP. McGraw-Hill, Inc., New York.

[63]. Zikopoulos, P., & Eaton, C. (2011). Understanding big data: Analytics for enterprise class Hadoop and streaming data. McGraw-Hill Osborne Media.

[64]. Marz, N., & Warren, J. (2015). Big data: Principles and best practices of scalable realtime data systems. Manning Publications Co.

[65]. Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2013). Applied multiple regression/correlation analysis for the behavioral sciences. Psychology press, Routledge.

[66]. Lin, W., Orgun, M.A., & Williams, G.J. (2002). An overview of temporal data mining. In: *Proceedings of the 1st Australian Data Mining Workshop*, 83-90.

[67]. Davenport, T.H., Harris, J.G., & Morison, R. (2010). Analytics at work: Smarter decisions, better results. Harvard Business Press, Buston, MA.

[68]. Trnka, A. (2014). Big data analysis. *European Journal of Science and Theology*, 10(1), 143-148.

[69]. Barras, C., Meignier, S., & Gauvain, J.-L. (2004). Unsupervised online adaptation for speaker verification over the telephone. In: *Proceedings of the IEEE Odyssey, ISCA Speaker Recognition Workshop*, 157-160.

[70]. Dowgiert, A. (2014). The impact of big data on traditional Health Information Management and EHR. Doctoral dissertation, The College of St. Scholastica.

[71]. Fotopoulou, E., Zafeiropoulos, A., Papaspyros, D., Hasapis, P., Tsiolis, G., Bouras, T., Zanetti, N. (2016). Linked data analytics in interdisciplinary studies: The health impact of air pollution in urban areas. *IEEE Access*, 4, 149-164.

[72]. Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6), 375-381.

[73]. Chapman, A.D. (2005). Principles and methods of data cleaning. *Report for the Global Biodiversity Information Facility*, Copenhagen, 1-72.

[74]. Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques, Elsevier.

[75]. Paunović, K. (2008). Data collecting. *Data Collecting Encyclopedia of Public Health*, Springer.196-199.

[76]. Ng, R.T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5), 1003-1016.

[77]. Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3), 264-323.

[78]. Han, J., & Cercone, N. (2000). RuleViz: a model for visualizing knowledge discovery process. In: *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 244-253.

[79]. Wnuk, A., Gozdowski, D., Górny, A., Wyszyński, Z., & Kozak, M. (2017). Data visualization in yield component analysis: an expert study. *Scientia Agricola*, 74(2), 118-126.

[80]. Imamura, T., Yamada, S., & Machida, M. (2011). Development of a high performance eigensolver on the petascale next generation supercomputer system. In: *Proceedings of the Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo*, 643-650.

[81]. Geng, L., & Hamilton, H.J. (2006). Interestingness measures for data mining: A survey. *ACM computing surveys*, 38(3), 1-32.

[82]. Chattratichat, J., Darlington, J., Ghanem, M., Guo, Y., Hüning, H., Köhler, M., Yang, D. (1997). Large scale data mining: challenges and responses. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Newport Beach, CA., 61-64.

[83]. Ren, S., & Zhao, X. (2015). A predictive maintenance method for products based on big data analysis. In: *Proceedings of the International Conference on Materials Engineering and Information Technology Applications*, 385-390.

[84]. Wu, X., Zhu, X., Wu, G.-Q., & Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), 97-107.

[85]. Bi, Z., & Cochran, D. (2014). Big data analytics with applications. *Journal of Management Analytics*, 1(4), 249-265.

[86]. Klösgen, W. (1996). Knowledge discovery in databases and data mining. *Foundations of Intelligent Systems*, 623-632.

[87]. Frigui, H. (2006). Membershipmap: Data transformation based on granulation and fuzzy membership aggregation. *IEEE Transactions on Fuzzy Systems*, 14(6), 885-896.

[88]. Blivband, Z., Grabov, P., & Nakar, O. (2004). Expanded Fmea (efmea). In: *Proceedings of the Annual Symposium Reliability and Maintainability*, 31-36.

[89]. Shafer, J., Agrawal, R., & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. In: *Proceedings of the 22nd International Conference on Very Large Databases*, Bombay, 544-555.

[90]. Luo, D., Ding, C., & Huang, H. (2012). Parallelization with multiplicative algorithms for big data mining. In: *Proceedings of the IEEE 12th International Conference on Data Mining*, 489-498.

[91]. Chen, R., Sivakumar, K., & Kargupta, H. (2004). Collective mining of Bayesian networks from distributed heterogeneous data. *Knowledge and Information Systems*, 6(2), 164-187.

[92]. Williams, P., Margules, C.R., & Hilbert, D.W. (2002). Data requirements and data sources for biodiversity priority area selection. *Journal of Biosciences*, 27(4), 327-338.

[93]. Maimon, O., & Rokach, L. (2005). Decomposition methodology for knowledge discovery and data mining. *Data Mining and Knowledge Discovery Handbook*, Springer, 2, 981-1003.

[94]. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A.H. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.

[95]. Duan, L., & Xiong, Y. (2015). Big data analytics and business analytics. *Journal of Management Analytics*, 2(1), 1-21.

[96]. Lohr, S. (2012). The age of big data. *New York Times* ,1-5.

[97]. Begoli, E., & Horey, J. (2012). Design principles for effective knowledge discovery from big data. In: *Proceedings of the Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, 215-218.

[98]. Chen, M., Mao, S., Zhang, Y., & Leung, V.C. (2014). Big data analysis. *Big Data*, Springer, 51-58.

[99]. Kurgan, L.A., & Musilek, P. (2006). A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, 21(01), 1-24.

[100]. Padmanabhan, B., & Tuzhilin, A. (1998). A brief-driven method for discovering unexpected patterns. In: *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, New York, 94-11.

[101]. Munir, S., Stankovic, J.A., Liang, C.-J.M., & Lin, S. (2014). Reducing energy waste for computers by human-in-the-loop control. *IEEE Transactions on Emerging Topics in Computing*, 2(4), 448-460.

[102]. Kane, M.J., Emerson, J., & Weston, S. (2013). Scalable strategies for computing with massive data. *Journal of Statistical Software*, 55(14), 1-19.

[103]. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11), 27-34.

[104]. Guarino, N. (1998). Formal ontology in information systems. In: *Proceedings of the 1st International Conference of Formal Ontology in Information Systems*, Trento, 46, 3-15.

[105]. Tempelaar, D.T., Rienties, B., & Giesbers, B. (2015). In search for the most informative data for feedback generation: Learning analytics in a data-rich context. *Computers in Human Behavior*, 47, 157-167.

[106]. Semantics (2016). Retrieved from http://www.historygraphicdesign.com/the-age-of-information/the-international-typographic-style/810-semantics.

[107]. Ramsay, S. (2003). Special Section: Reconceiving text analysis: Toward an algorithmic criticism. *Literary and Linguistic Computing*, 18(2), 167-174.

[108]. Lenz, H.-J. (1994). A rigorous treatment of microdata, macrodata, and metadata. In: *Proceedings of the Computational Statistics Conference*, Heidelberg, 357-362.

[109]. Berry, S., Levinsohn, J., & Pakes, A. (2004). Differentiated products demand systems from a combination of micro and macro data: The new car market. *Journal of Political Economy*, 112(1), 68-105.

[110]. Icke, I., & Rosenberg, A. (2010). Multi-objective genetic programming projection pursuit for exploratory data modeling. In: *Proceedings of the 5th Annual Machine Learning Symposium*, New York, x-x.

[111]. Shehada, M., & Alkhaldi, F. (2015). Measuring the efficiency and effectiveness of the human resources training function at Orange Jordan. *International Journal of Quantitative and Qualitative Research Method*, 3(2), 1-12.

[112]. Parzen, E. (1979). Nonparametric statistical data modeling. *Journal of the American Statistical Association*, 74(365), 105-121.

[113]. Michener, W.K., & Jones, M.B. (2012). Ecoinformatics: Supporting ecology as a data-intensive science. *Trends in Ecology & Evolution*, 27(2), 85-93.

[114]. Gorissen, D., Crombecq, K., Hendrickx, W., & Dhaene, T. (2006). Grid enabled metamodeling. In: *Proceedings of the 7th International Meeting on High Performance Computing for Computational Science*. Retrieved from: http://sumo.intec.ugent.be/sites/sumo/files/sumo/2006_07__VECPAR.pdf.

[115]. Gorissen, D., Crombecq, K., Couckuyt, I., & Dhaene, T. (2009). Automatic approximation of expensive functions with active learning. *Foundations of Computational Intelligence, Learning and Approximation*, Springer, 1, 35-62.

[116]. Nemchinova, Y., & Sayani, H. (2006). Using tools in teaching university courses in information technology. In: *Proceedings of the 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 361-367.

[117]. Lenzi, S.E., Standoli, C.E., Andreoni, G., Perego, P., & Lopomo, N.F. (2018). A software toolbox to improve time-efficiency and reliability of an observational risk assessment method. In: *Proceedings of the Congress of the International Ergonomics Association*, 689-708.

[118]. Murakoshi, H., Kishi, M., & Ochimizu, K. (2002). Developing web-based on-demand learning system. In: *Proceedings of the International Conference on Computers in Education*, 1223-1227.

[119]. Sain, J.A. (1986). ESD/MITRE Software Acquisition Symposium Proceedings; an ESD/Industry Dialogue held in Bedford, Massachusetts on May 6-7, 1986. Retrieved from: http://www.dtic.mil/docs/citations/ADA178785.

[120]. Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National Science Review*, 1(2), 293-314.

[121]. Roy, R., Shehab, E., Tiwari, A., Baines, T., Lightfoot, H., Benedettini, O., & Kay, J. (2009). The servitization of manufacturing: A review of literature and reflection on future challenges. *Journal of Manufacturing Technology Management*, 20(5), 547-567.

[122]. Oussous, A., Benjelloun, F.Z., Lahcen, A.A, & Belfkih, S. (2018).Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4), 431-448.

[123]. Taneja, A. (2017). Enhancing web data mining: The Study of factor analysis. *Web Usage Mining Techniques and Applications Across Industries*, IGI Global, 116-136.

[124]. Wang, Y., Kung, L., & Byrd, T.A. (2018). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, 126, 3-13.

[125]. Mayne, B.T., Leemaqz, S.Y., Buckberry, S., Lopez, C.M.R., Roberts, C.T., Bianco-Miotto, T., & Breen, J. (2018). msgbsR: An R package for analysing methylation-sensitive restriction enzyme sequencing data. *Scientific reports*, 8(1), 2190-2198.

[126]. Tremsin, A.S., Ganguly, S., Meco, S.M., Pardal, G.R., Shinohara, T., & Feller, W.B. (2016). Investigation of dissimilar metal welds by energy-resolved neutron imaging. *Journal of Applied Crystallography*, 49(4), 1130-1140.

[127]. Roe, D.R., & Cheatham III, T.E (2018). Parallelization of CPPTRAJ enables large scale analysis of molecular dynamics trajectory data. *Journal of Computational Chemistry*, 39(25), 2110-2117.

[128]. Boettcher, A., Brendel, W., & Bethge, M. (2016). Large scale blind source separation. In: *Proceedings of Bernstein Conference*, 118-119.

[129]. Thilagaraj, T., & Sengottaiyan, N. (2017). A review of educational data mining in higher education system. In: *Proceedings of the 2nd International Conference on Research in Intelligent and Computing in Engineering*, 349-358.

[130]. Spatti, D.H., & Liboni, L.H. (2016). Computational tools for data processing in smart cities. *Smart Cities Technologies*, InTech, 41-54.

[131]. Rizwan, A., Zoha, A., Zhang, R., Ahmad, W., Arshad, K., Ali, N.A, ..., & Abbasi, Q.H. (2018). A review on the role of nano-communication in future healthcare systems: A big data analytics perspective. *IEEE Access*, 6, 41903-41920.

[132]. Tian, Y. (2017). Accelerating data preparation for big data analytics. Doctoral dissertation, Télécom ParisTech.

[133]. Demuth, H., Beale, M., & Hagan, M. (2008). Neural network toolbox 6. *User's guide*, 37-55.

[134]. Foster, E.C., & Godbole, S. (2016). Overview of Oracle. *Database Systems*, Apress, Berkeley, CA, 435-442.

[135]. Vijayaraj, J., Saravanan, R., Paul, P.V., & Raju, R. (2016). Hadoop security models-a study. In: *Proceedings of the Online International Conference on Green Engineering and Technologies*, 1-5.

[136]. Sahu, S.K., Jacintha, M.M., & Singh, A.P. (2017). Comparative study of tools for big data analytics: An analytical study. In: *Proceedings of the International Conference on Computing, Communication and Automation*, IEEE, 37-41.

[137]. Piatetsky, G. (2016). R, Python Duel as top analytics, data science software. *Kdnuggets 2016 Software Poll results*. Retrieved from: http://www.kdnuggets.com/2016/06/r-python-topanalytics-data-mining-data-science-software.html.

[138]. Ye, F., Chen, Y., Huang, Q., & Li, L. (2017). Developing cloud-based tools for water resources data analysis using R and Shiny. In: *Proceedings of the International Conference on Emerging Internetworking, Data & Web Technologies*, 289-297.

[139]. Goyal, A., Khandelwal, I., Anand, R., Srivastava, A., & Swarnalatha, P. (2016). A comparative analysis of the different data mining tools by using supervised learning algorithms. In: *Proceedings of the International Conference on Soft Computing and Pattern Recognition*, 105-112.

[140]. Giraud-Carrier, C., & Povel, O. (2003). Characterising data mining software. *Intelligent Data Analysis*, 7(3), 181-192.

[141]. Scarnò, M. (2008). ADaMSoft: un software open source e un'esperienza nel calcolo scientifico. *CASPUR Annual Report*, 46-47.

[142]. Deng, H.W., Zhang, Y.N., Ke, B., & Li, M.T (2017). Decision making of ore mining model based on analytica software. *The Chinese Journal of Nonferrous*

Metals. Retrieved from:http://en.cnki.com.cn/Article_en/CJFDTOTAL-ZYXZ201702014.htm.

[143]. Cieplik, U. (2006). BV4. 1 Methodology and user-friendly software for decomposing economic time series. German Federal Statistical Office.

[144]. Karypis, G. (2002). CLUTO-a clustering toolkit. Technical Report 02-017. Department of Computer Science, University of Minnesota. Retrieved from http://www.cs.umn.edu/cluto.

[145]. Comsol, A.B. (2016). COMSOL Multiphysics® v. 5.2. Stockholm, Sweden.

[146]. Landau, I.D., & Landau, V. (2016). Data mining et machine learning dans les big data: une tentative de démystification. Retrieved from: https://hal.archives-ouvertes.fr/hal-01393640/file/DMML_F.pdf.

[147]. Chekanov, S.V. (2016). Numeric computation and statistical data analysis on the Java platform. Springer, Basel.

[148]. Coman, E., Brewster, M.W., Popuri, S.K., Raim, A.M., & Gobbert, M.K. (2012). A comparative evaluation of Matlab, Octave, FreeMat, Scilab, R, and IDL on Tara. Retrieved from: http://www.webcitation.org/6BbWqerg3.

[149]. Eaton, J.W. (2017). GNU Octave 4.2 Reference Manual. Samurai Media Limited.

[150]. JASP Team (2016). JASP (Version 0.7. 5.5) [Computer software]. *Google Scholar*, 765-766.

[151]. Dwivedi, S., Kasliwal, P., & Soni, S. (2016). Comprehensive study of data analytics tools (RapidMiner, Weka, R tool, Knime). In: *Proceedings of the Symposium on Colossal Data Analysis and Networking*, 1-8.

[152]. Higham, D.J., & Higham, N.J. (2016), MATLAB Guide. Siam.

[153]. Hothorn, T. (2007). Maxstat: Maximally selected rank statistics. R package version 0.7 14. Retrieved from http://cran/.R-project.org/package=maxstat.

[154]. Winston, W. (2016). Microsoft Excel Data Analysis and Business Modeling. Microsoft Press.

[155]. Miller, W. (2012). OpenStat reference manual. Springer, Science & Business Media.

[156]. Tamayo, P., Berger, C., Campos, M., Yarmus, J., Milenova, B., Mozes, A., Thomas, S. (2005). Oracle data mining. *Data Mining and Knowledge Discovery Handbook*, 1315-1329.

[157]. Shmueli, G., Bruce, P.C., Yahav, I., Patel, N.R., & Lichtendahl Jr, K.C. (2017). Data mining for business analytics: concepts, techniques, and applications in R. John Wiley & Sons.

[158]. Hofmann, M., & Klinkenberg, R. (2016). Getting Used to RapidMiner Ingo Mierswa. *RapidMiner*, Chapman and Hall/CRC, 73-84.

[159]. Cody, R. (2018). Learning SAS by example: a programmer's guide. SAS Institute.

[160]. Campbell, S.L., Chancelier, J.P., & Nikoukhah, R. (2006). Introduction to Scilab. *Modeling and Simulation in Scilab/Scicos*, 9-71.

[161]. Sonnenburg, S., Henschel, S., Widmer, C., Behr, J., Zien, A., Bona, F. D., Franc, V. (2010). The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11(Jun), 1799-1802.

[162]. George, D., & Mallery, P. (2016). IBM SPSS statistics 23 step by step: A simple guide and reference. Routledge.

[163]. StataCorp, L. (2007). Stata data analysis and statistical Software. *Special Edition Release*, 10, 733.

[164]. Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2016). WEKA manual for version 3-9-1. The University of Waikato, Hamilton, New Zealand.

[165]. Peng, S., Wang, G., & Xie, D. (2016). Social influence analysis in social networking big data: Opportunities and challenges. *IEEE network*, 31(1), 11-17.

[166]. Taleb, I., Serhani, M.A., & Dssouli, R. (2018). Big data quality: A survey. In: *Proceedings of the International Congress on Big Data*, IEEE, 166-173.

[167]. Gepp, A., Linnenluecke, M.K., O'Neill, T.J., & Smith, T. (2018). Big data techniques in auditing research and practice: Current trends and future opportunities. *Journal of Accounting Literature*, 40, 102-115.

[168]. Prakash, B., & Hanumanthappa, D.M. (2014). Issues and challenges in the era of big data mining. *International Journal of Emerging Trends and Technology in Computer Science*, 3, 321-325.

[169]. Vijayalakshmi, B., & Srinath, M. (2014). State-of-the-art frameworks and platforms for processing and managing big data as well as the efforts expected on big data mining. *Elysium Journal of Engineering Research & Management*, 1(2), 1-9.

[170]. Joshi, S., & Nair, M.K. (2018). Survey of classification based prediction techniques in healthcare. *Indian Journal of Science and Technology*. Retrieved from: http://www.indjst.org/index.php/indjst/article/viewFile/121111/84191.

[171]. Brennan, R. (2017). Challenges for value-driven semantic data quality management. In: *Proceedings of the 19th International Conference on Enterprise Information Systems*, 1, 385-392.

[172]. Rahimah, A.A., & Naik, T.R. (2015). Data mining for big data. *International Journal of Advanced Technology and Innovation research*, 7(5), 697-701.

[173]. Bikakis, N. (2018). Big data visualization tools. Encyclopedia of Big Data Technologies.

[174]. Chen, M., Mao, S., Zhang, Y., & Leung, V.C. (2014). Big data applications. *Big Data*, Springer, 59-79.

[175]. Kopanas, I., Avouris, N.M., & Daskalaki, S. (2002). The role of domain knowledge in a large scale data mining project. In: *Proceedings of the 2nd*

Hellenic Conference on Artificial Intelligence: Methods and Applications of Artificial Intelligence, 288-299.

[176]. Brodie, M., Greaves, M., & Hendler, J. (2011). Databases and AI: The twain just met. In: *Proceedings of the STI Semantic Summit*, Riga, Latvia, 6-8.

[177]. Bizer, C., Boncz, P., Brodie, M.L., & Erling, O. (2012). The meaningful use of big data: four perspectives - Four challenges. *ACM Sigmod Record*, 40(4), 56-60.

[178]. Michael, K., & Miller, K.W. (2013). Big data: New opportunities and new challenges [guest editors' introduction]. *Computer*, 46(6), 22-24.

[179]. Schöner, H.-P. (2004). Automotive mechatronics. *Control engineering practice*, 12(11), 1343-1351.

[180]. Zheng, C., Bricogne, M., Le Duigou, J., & Eynard, B. (2014). Survey on mechatronic engineering: A focus on design methods and product models. *Advanced Engineering Informatics*, 28(3), 241-257.

[181]. Yang, X., Moore, P., & Chong, S.K. (2009). Intelligent products: From lifecycle data acquisition to enabling product-related services. *Computers in Industry*, 60(3), 184-194.

[182]. Breuer, P., Moulton, J., & Turtle, R.M. (2013). Applying advanced analytics in consumer companies. *McKinsey & Company*, New York.

[183]. Li, J., Tao, F., Cheng, Y., & Zhao, L. (2015). Big data in product lifecycle management. *The International Journal of Advanced Manufacturing Technology*, 81(1-4), 667-684.

[184]. Shaw, M.J., Subramaniam, C., Tan, G.W., & Welge, M.E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems*, 31(1), 127-137.

[185]. Duncan, G. (2007). Privacy by design. *Science*, 317(5842), 1178-1179.

[186]. Schadt, E.E. (2012). The changing privacy landscape in the era of big data. *Molecular Systems Biology*, 8(1), 612.

[187]. Liu, C., & Arnett, K.P. (2000). Exploring the factors associated with Web site success in the context of electronic commerce. *Information & Management*, 38(1), 23-33.

[188]. Preis, T., Moat, H.S., & Stanley, H.E. (2013). Quantifying trading behavior in financial markets using Google Trends. *Scientific reports*, 3, 1684.

[189]. Fuhrer, E. (2000). System for enhanced financial trading support. Google Patents, Washington, DC.

[190]. Bryant, R., Katz, R.H., & Lazowska, E.D. (2008). Big-data computing: Creating revolutionary breakthroughs in commerce, science and society. Retrieved from: http://www.cra.org/ccc/files/docs/init/Big_Data.pdf,1-15.

[191]. Steinbrook, R. (2008). Personally controlled online health data – the next big thing in medical care?. *The New England Journal of Medicine*, 358(16), 1653.

[192]. Groves, P., Kayyali, B., Knott, D., & Van Kuiken, S. (2013). The big data revolution in healthcare. *McKinsey Q*. Retrieved from: http://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/the-big-data-revolution-in-us-health-care.

[193]. Organization, W.H. (2007). WHO operational package for assessing, monitoring and evaluating country pharmaceutical situations. Guide for Coordinators and Data Collectors.

[194]. Weiss, G.M. (2005). Data mining in telecommunications. *Data Mining and Knowledge Discovery Handbook*, 1189-1201.

[195]. Kļevecka, I., & Lelis, J. (2008). Pre-processing of input data of neural networks: the case of forecasting telecommunication network traffic. *Telektronikk*, 3, 168-178.

[196]. Szalay, A. (2011). Extreme data-intensive scientific computing. *Computing in Science & Engineering*, 13(6), 34-41.

[197]. Brumfiel, G. (2011). Down the petabyte highway. *Nature News*, 469(20), 282-283.

[198]. Wang, F.-Y., Carley, K.M., Zeng, D., & Mao, W. (2007). Social computing: From social informatics to social intelligence. *IEEE Intelligent Systems*, 22(2), 79-83.

[199]. Lesk, A. (2013). Introduction to bioinformatics. Oxford University Press, 3rd edition.

[200]. McDermott, J., Samudrala, R., Bumgarner, R., Montgomery, K., & Ireton, R. (2009). Computational systems biology. Springer.

[201]. Chen, C.P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314-347.

[202]. Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. *ACM SigKDD Explorations Newsletter*, 14(2), 1-5.

[203]. Leskovec, J., Rajaraman, A., & Ullman, J.D. (2014). Mining of massive datasets. Cambridge University Press.

[204]. Zheng, M., Ming, X., Li, G., & Shi, Y. (2016). The framework of business model innovation for smart product-service ecosystem. In: *Proceedings of the NordDesign Conference*, 2, 400-409.

[205]. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15), 2787-2805.

[206]. Dillon, T.S., Zhuge, H., Wu, C., Singh, J., & Chang, E. (2011). Web-of-things framework for cyber-physical systems. *Concurrency and Computation: Practice and Experience*, 23(9), 905-923.

[207]. Farringdon, J., Moore, A.J., Tilbury, N., Church, J., & Biemond, P.D. (1999). Wearable sensor badge and sensor jacket for context awareness. In: *Proceedings*

of the 3rd International Symposium on Wearable Computers, San Francisco, CA., 107-113.

[208]. Valckenaers, P., Saint Germain, B., Verstraete, P., Van Belle, J., Hadeli, K., Van Brussel, H. (2009). Intelligent products: agere versus essere. *Computers in Industry*, 60(3), 217–228.

[209]. Horváth, I., Rusák, Z., & Li, Y. (2017). Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems. In: *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 1-14.

[210]. Meyer, G.G., Främling, K., & Holmström, J. (2009). Intelligent products: A survey. *Computers in Industry*, 60(3), 137-148.

[211]. Främling, K., Holmström, J., Loukkola, J., Nyman, J., & Kaustell, A. (2013). Sustainable PLM through intelligent products. *Engineering Applications of Artificial Intelligence*, 26(2), 789-799.

[212]. Earley, S. (2015). Analytics, machine learning, and the internet of things. *IT Professional*, 17(1), 10-13.

[213]. Szykman, S., Fenves, S.J., Keirouz, W., & Shooter, S.B. (2001). A foundation for interoperability in next-generation product development systems. *Computer-Aided Design*, 33(7), 545-559.

[214]. Aroyo, L., & Dicheva, D. (2004). The new challenges for e-learning: The educational semantic web. *Educational Technology & Society*, 7(4), 59-69.

[215]. Bézivin, J., Bruneliere, H., Jouault, F., & Kurtev, I. (2005). Model engineering support for tool interoperability. In: *Proceedings of the Workshop in Software Model Engineering - A MODELS 2005 Satellite Event*, Montego Bay, Jamaica, 1-16.

[216]. Pandey, S., & Nepal, S. (2013). Cloud computing and scientific applications - big data, scalable analytics, and beyond. *Future Generation Computer Systems*, 29, 1774-1776.

[217]. Haugen, H., & Ask, B. (2010). Real integration of ICT into subject content and methodology requires more than technology, infrastructure and standard software. In: *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 107-117.

[218]. Nabati, E.G., Thoben, K.D., & Daudi, M. (2017). Stakeholders in the middle of life of complex products: understanding the role and information needs. *International Journal of Product Lifecycle Management*, 10(3), 231-257.

[219]. van der Vegte, W.F. (2016). Taking advantage of data generated by products: trends, opportunities and challenges. In: *Proceedings of the International ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Charlotte, USA, x-x.

[220]. Chow, C.W., Liu, J., Li, J., Swain, N., Reid, K., & Saint, C.P. (2018). Development of smart data analytics tools to support wastewater treatment plant operation. *Chemometrics and Intelligent Laboratory Systems*, 177, 140-150.

Chapter 3

Research cycle 2: Building a qualitative theory and framework of the needs

3.1. Introduction

3.1.1. Objectives and activities of the second research cycle

The second research cycle focused on building a set of fundamentals and requirements needed to realize a next generation data analytics toolbox. The outcomes of the literature study were the entry point for the research activities presented here. Based on the primary findings and the gaps identified in the literature, we planned two studies for this chapter. First, we organized and conducted a web-based interrogation to understand designers' practical needs and opinions related to data analytics practices and tools. This was then complemented by the synthesis of theories needed for DAT development. This synthesis merged theories using the methodology of ATF developed for this concrete purpose.

First, the QBI explored the needs associated with the daily data analytics and product enhancement practices of white goods developers and the potential for fulfilling these needs based on current knowledge and massive data methodologies and technologies. The outcomes of the QBI were compared to the literature to reflect on the reality and feasibility of the needs and, correspondingly, to associate the needs with possible new functionalities and services of next generation data analytics tools. Second, ATF was crystallized and used in the practical development of the new data analytics toolbox (or tools). Data-driven product enhancement by white goods designers was considered as the specific application perspective. To this end, five theories were fused: (i) a theory concerning designers' needs (built from the QBI), (ii) a theory describing advanced technological enables, (iii) a theory explaining the evolution of data analytics, (iv) a theory of combined creative problem-solving and decision-making, and (v) a theory of functional and structural interoperability. These theories were subject to all steps of ATF, and their merging allowed the creation of a new theory as the basis of a next generation SDATB. In this chapter, the logic, components, and processes of both QBI and ATF are presented. The outcomes are a set of fundamentals, requirements, and functions needed for building a next generation SDATB.

3.1.2. Methodology applied in the second research cycle

The research objectives of this chapter led to two topics being considered in this second research cycle, one inductive and one deductive. The inductive study was steered by the principle of bottom-up knowledge aggregation, the deductive study by that of top-down argumentation (logical reasoning). From a methodological point of view, the inductive study was a web-hosted QBI with the specific objective of synthesizing an explanatory theory in a bottom-up manner. The deductive study involved axiomatization-based conceptual discretization of relevant theories as well as semantic fusion of axioms and supplementary postulates into the body of a new and synthetic explanatory theory.

The methodological framing applied in the first topic was RDC. Based on its principles, two successive phases took place, namely an explorative and a confirmative phase. The exploration consisted of a QBI and a literature study, first to investigate designers' needs from a practical point of view and, later, to derive knowledge from a state-of-the art point of view. The outcomes of the two studies were compared in the confirmative phase to identify matches and mismatches and consolidate the results accordingly. For the second topic, RDC was again the applied methodological framing. The explorative phase consisted of investigating existing theories related to the development of data analytics tools and building the methodology to combine them in this specific context. The confirmative phase applied this methodology to create a new theory consolidating requirements, principles, and functionalities of a new SDATB that considers practical designers' needs.

3.2. Inductive study: Web-hosted questionnaire-based interrogation

3.2.1. Setting the stage

The overall picture of competitive product development and innovation is made more complicated by the emergence and manifestation of the capability of smartness in products [1]. The proliferation of smart products forces companies to rethink and retool almost everything they do internally [2]. Such products incorporate various self-learning, self-adaptation, and self-management capabilities [3]. They may actively generate, collect, and communicate a large amount of data about their operational states and use circumstances, and they can reason with these data [4]. Still, substantial research effort needs to be invested in this rapidly expanding field of interest. The main issues for studies include the following: (i) how can (system) smartness be self-managed and self-organized, (ii) how can it be utilized in function and performance enhancement, (iii) how can it be implemented in existing products and services, and (iv) how can new generations of smart products and services be brought to existence.

The activity in this section was driven by the following research question:

> *What do designers miss related to the use and outputs of current data analytics tools in the context of possible product improvements?*

In the context of this question, we refer to the review of the state of the art presented in Chapter 2, which found that the currently available tools, for instance, do not consider the changing aspects of design tasks and do not provide correlated output data structures. Consequently, our working hypothesis was formulated as follows:

> *Interviewing different product designers about their daily design activities related to a particular category of products and about their experiences using data analytics tools in the context of product improvement will help determine their needs and will provide insights on what they miss in existing software tools.*

In operationalizing the research, we encountered two challenges. One of these was rooted in the fact that the study could not be conducted and meaningful results could not be expected without focusing on a specific family of products. A finding that is valid for a given family of products may be irrelevant or even incorrect for a different family of products. Therefore, we restricted our investigations to a relatively broad but also widely used family of white goods. The second challenge was that it was not possible to aggregate an all-encompassing body of knowledge about proliferation, functionalities, applicability, industrial uses, best practices, and use experiences of DATs in the context of the above family of products. For this reason, we restricted ourselves to the identification of those tools that had strong footprints in the domain of white goods, and we collected operational and application information only about these.

Since at the start of our research project the topic of using data analytics tools in the daily product development practice of small and medium-sized enterprises was in an emerging stage, we faced some technical challenges as well. For instance, (i) the related literature was limited, (ii) distribution of the related specific publications was uneven both chronologically and with respect to content, and (iii) factual data concerning the use of data analytics tools in design contexts were scarce. Another technical challenge concerned sampling the subjects for the inquiries, since each of them involved a different type of product. Yet another technical challenge was compiling a full-fledged, exploratory, and clearly worded questionnaire.

Finally, our guiding research question also carried a hidden complexity because it brought three issues into interrelationship. These issues were (i) the commercial choice and industrial use of current data analytics tools, (ii) the information and knowledge that could be generated by these tools when used by designers not expert in data analytics, and (iii) the exploration and/or inception of improvement opportunities for the target product family.

3.2.2. Basic considerations for the web-hosted interrogation

Our exploratory literature survey cast light on the issue of additional functionalities and services that product designers would expect from advanced DATs, beyond what is offered by current commercial and academic tools. With these in mind, we made assumptions about important aspects of the research phenomenon:

- Development of white goods covers a broad field of product development. These products are already equipped with sophisticated control units, and their evolution continues towards smart and autonomous products.
- The MoL stage is the longest period in the product lifecycle. The data collected in this period can provide a large amount of information about the use, servicing, and maintenance of products, which can be extremely useful for product improvement

and innovation.

- The marketed data analytics tools have been developed based on the needs, knowledge, and skills of data experts and software engineers rather than those of product designers.
- It is believed that designers need different function and interfaces than offered by many current software tools and packages.

The forerunning survey also triggered a number of working research questions, such as the following:

- Do MoLD get proper attention from product designers?
- What are the data transformation steps included in typical design processes?
- Are product designers satisfied with the quality of the data they use in their daily product development activities?
- How and for what purposes do designers use data analytics tools in designing white goods?
- What are product designers' professional experiences with data analytics tools?
- What do product designers most like or dislike about the data analytics tools they use?

Driven by these specific research questions, we focused on (i) the activities of product designers, (ii) the exploitation of MoLD, (iii) the use of data analytics tools, and (iv) the exploration of product improvement opportunities. We also devised a generic working scenario for product developers:

- Step 1: For the targeted family of products, product developers (not specialized in data analytics) generate data, information, and knowledge about the MoL stage of products by using advanced data analytics tools, technologies, and assets.
- Step 2: They convert the MoLD, information, and knowledge into generic product enhancement options and concrete concepts.
- Step 3*:* They operationalize ideas, concepts, and analogies in the framework of a strategy plan or vision and/or in concrete product enhancements and new design options.
- Step 4: They make decisions about the changes to be introduced with the goal of optimally enhancing elements or the whole of the product family.

In developing our questionnaire, we used this four-step scenario as a high-level (generic) logical framework of the activities of product designers.

3.2.3. Framing the web-hosted questionnaire-based interrogation

The specific objective of the interrogative study was to determine (i) what data analytics approaches and tools the participants used and for what purposes, (ii) which of these they found useful in the specific application field, (iii) what their expectations were for data analytics tools, and (iv) what other technical support they needed. As mentioned earlier, the target application field was enhancement and new development of white goods. This family of products mainly includes household appliances such as refrigeration equipment, different types of cookers, microwave ovens, washing equipment, drying equipment, air conditioners, and so on. Based on the outcome of the interrogative study, we intended to construct an explanatory theory about the relationship of data analytics tools and product designers and developers. We planned to synthesize the theory by generalizing from the specific situations and opinions of the interviewees.

The interrogative study was intended to extract knowledge about multiple specific concerns. We wanted to know when in the design process designers used data collected or generated by products. We were curious about how they collected and processed the data with their available data analytics tools. We were also interested in finding information about the needs and preferences of designers regarding the outputs of data analytics tools. We wanted to use this intelligence to feed a solution proposal that could better meet their expectations by including new functionalities, affordances, and approaches to future (and data-intensive) generations of white goods.

As a first activity in developing the questionnaire, we considered its overall structure. The structuring was guided partly by the specific research questions and partly by the identification of those chunks of information that were deemed necessary for the study. These were (i) the description of the application field, (ii) the objectives of the interrogation, (iii) the participants we wanted to focus on, (iv) the knowledge we aimed to extract, and (v) the concrete research questions to be answered. We specified the approach of the interrogation by (i) choosing the proper questionnaire-based interviewing method, (ii) specifying an adequate sampling method, (iii) calculating the minimum sample size based on the research variables and bias and error assumptions, and finally (iv) defining the detailed conduct of the interrogation. We followed this by specifying the contents of the questionnaire, which included (i) logical decomposition of the named question categories, (ii) formulation of the individual questions, and (iii) specification of the order and semantic relationships between questions.

3.2.4. Piloting and sampling

Before conducting the online interrogative study, we pretested the draft questionnaire in a preliminary test called the “pilot experiment.” The single objective of the pilot was to improve the structure of the questionnaire and the comprehensibility of the questions. Therefore, we evaluated the following aspects of the content: (i) comprehensiveness, (ii) feasibility, (iii) usefulness, (iv) time aspects, and (v) internal consistency. Table 3.1 summarizes the objectives and the major findings of the pilot study as well as the changes introduced in the questionnaire to improve its quality.

The first block in Table 3.1 includes the objectives of the pilot experiment, expressed in the form of questions. The second block lists the most significant observations based on the participants’ answers. The third block describes the improvements introduced in the questionnaire based on the outcomes of the pilot.

Completing the pilot helped us restructure the questions in a more direct and consistent form. We were able to evaluate (i) the subjects' understanding of the questions, (ii) the appropriateness of the questions order, (iii) the utility of the questions given the interrogation's goals, (iv) the sufficiency of the knowledge obtained, and (v) the time required to complete the questionnaire. Based on the observations and evaluations, we

Table 3.1. Piloting objectives, findings, and improvements

Objectives	- Was the time duration of the questionnaire realistic? - Were the participants willing to answer nonmandatory questions? - Did the participants give proper answers to the questions? - Were there overlapping questions from the participants' point of view? - Were the answers obtained in line with the objectives of the interrogation? - Was the obtained knowledge enough to make?
Findings	- The time duration mentioned in the introduction of the questionnaire was not realistic - The majority of participants did not answer all the questions even if they were important for the study - A few questions were not needed - Some questions were not understood by the participants - Some questions were too abstract and needed some further explanation - More options were needed to be given as answers to questions by the participants
Improvements	- Extra 5 minutes were added to the answering time - The introduction of the survey was restructured - Some questions were removed - Logical links between the questions were added (the possibility of skipping questions was introduced if the answer was no to a given question) - All multiple choice questions were made obligatory to answer - The ordering of some questions was changed - The structure and sentence construction of the questions was changed - More options were added to multiple choice questions - The option "I do not know" was added - Group two questions or more into one question - Explanations were added concerning some used technical words and sentences

made many modifications before consolidating the final version of the questionnaire and sending it to all targeted participants. Those modifications included enhancing the interrogation structure, the questions, and the flow, as well as changing the sequence of the questions.

To achieve a high level of reliability in our knowledge exploration, we had to determine the sample size carefully. The strategy of sampling for the full-scale study was (i) to create a representative sample of the population considering all features and all important aspects, and (ii) to provide a robust statistical basis for generalizing the results obtained with the sample to the assumed whole population of white goods designers. We considered a sample size that is typical for descriptive studies [5].

An appropriate sample size for descriptive studies generally depends on the following parameters: (i) minimum expected difference (MED), (ii) the estimated measurement variability, and (iii) the significance criterion factor. In our case, we chose 10% (0.10) for MED, considering it acceptable for our interrogation procedure to be 80% accurate. The value of the significance criterion factor was chosen to be 0.05 (that is the probability associated with the corresponding 95% confidence interval). Based on this, the value of z_{crit} is 1.96. The value of the estimated measurement variability was taken as the value of the standard deviation (δ) in the responses and was subjectively expected to be lower than 15% (0.15). The numerical calculation was made using the following equation:

$$N' = \frac{4\delta^2(z_{crit})}{MED^2}. \tag{3.1}$$

This resulted in a value $N' = 17.64$, which was rounded up to $N = 18$ participants. Here we need to consider the acceptable response rate "η." Normally, a response rate of 20% ($\eta = 0.2$) is considered a good result. However, given the uncertainties in our estimate $\eta = 0.15$. It means that $P = N/\eta$ people should be invited as respondents. Numerically, this means approximately 120 people. We contacted this number of people to obtain a sufficient amount of statistical data.

3.2.5. Content of the web-hosted questionnaire-based interrogation

The questionnaire was the same for all invited participants. The scheme of the questionnaire is shown in Figure 3.1. It consisted of six sections: (i) subject information, (ii) product and task information, (iii) knowing and using big data, (iv) knowing and using data analytics tools, (v) needed services and functionalities, and (vi) additional information and suggestions. The various sections contained purposeful combinations of (i) direct questions, (ii) multiple-choice questions, (iii) pointed questions, (iv) evaluative questions, and (v) open-ended questions. The first two sections of the questionnaire were informative in nature. The subject information section was divided into two blocks, named (i) knowing the participants and (ii) company interest. Six questions were included in the former block, asking about (i) the level of participants' education, (ii) their years of experience in design, (iii) the years spent in their current institution, and (iv) the years spent in their current job and position.

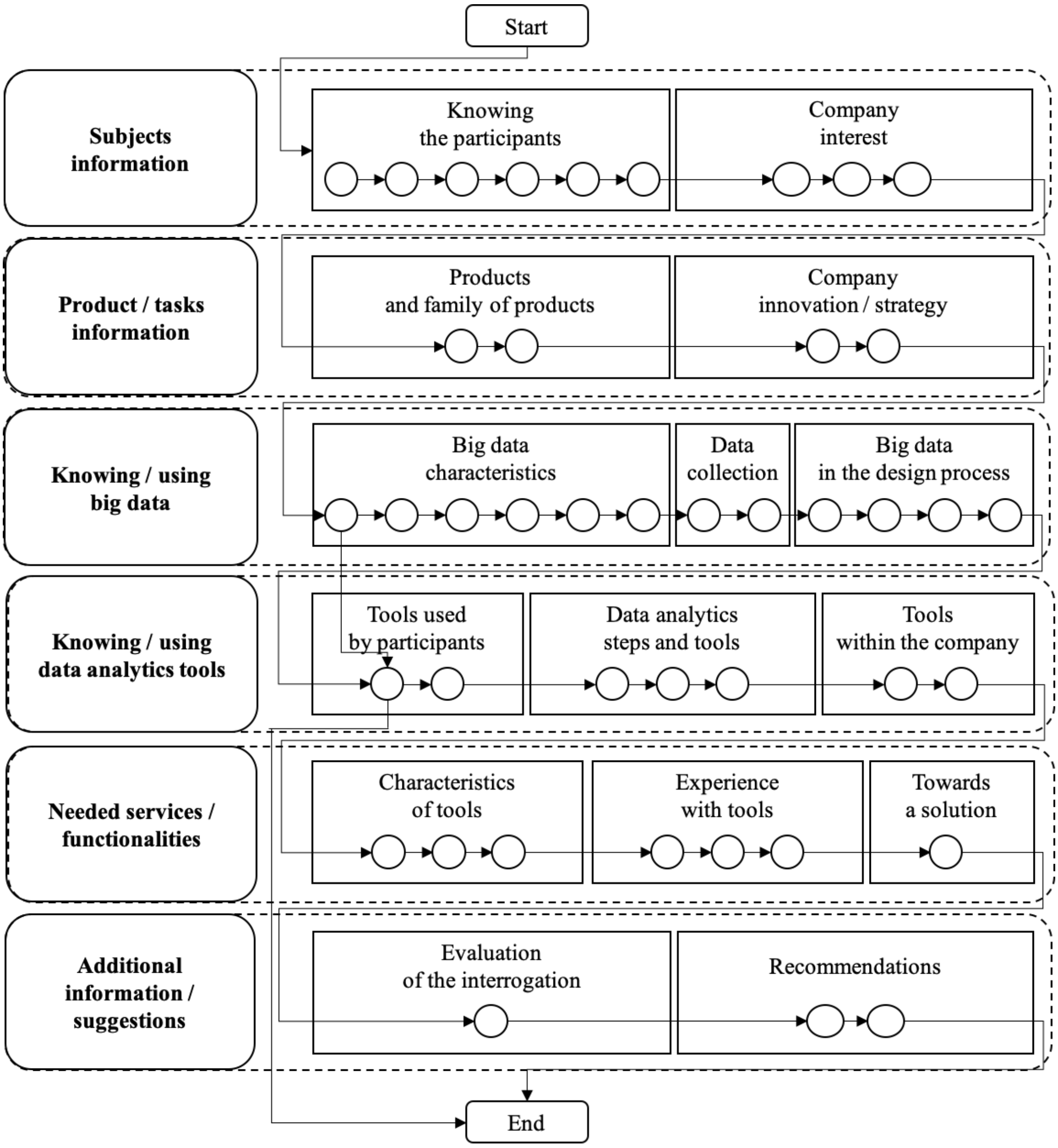


Figure 3.1. Flow of the conduct of the web-hosted questionnaire-based interrogation

The questions in the latter block were about (i) the company's interest in using data in daily design activities, (ii) the company's job locations, and (iii) the countries targeted by the designed products. The second section collected information about the designed products and the design tasks. This section was divided into two blocks: (i) family and kinds of products and (ii) the innovation strategies of the company. The former block contained two questions about the family of products the participant was involved in designing, developing, and/or managing and about the specific products they had personally designed. The questions in the second block asked about the innovations the company supported and about its business strategies. The responses were consolidated by considering the participants' profiles and their design experience.

The third section investigated the use of MoLD in design processes. We wanted to learn whether this issue is getting proper attention within companies. We also wanted to determine whether there were crucial data transformation steps included in the current design activities and whether the designers found the quality of the collected data satisfactory. This section was divided into three blocks that focused on (i) the characteristics of big data, (ii) the approaches to data collection, and (iii) the use of big data in the design process. The questions included in the first block asked the participants about (i) their use of big data in their daily design tasks, (ii) the sources of data, (iii) the volume of data, (iv) the type of data they used, (v) the nature of their raw data, and (vi) the time dimension of the data. The questions in the second block inquired about (i) the stages of data collection and (ii) the participants' satisfaction with the data contents and quality. The questions in the third block were formulated to explore (i) the tasks big data were used for in the design process, (ii) the data transformation steps executed before data were used in the design process, and (iii) the defined criteria for characterizing the data as valuable or useful.

The fourth section of the questionnaire featured three blocks that covered (i) the use of data analytics tools by product designers, (ii) the difficulties they encountered using them and their level of satisfaction with the tools' functionalities, and (iii) opportunities offered by the company to use data analytics tools. The first block contained questions about (i) whether the participants requested and used data analytics tools and software packages in their product development tasks and (ii) about which tools they had experience with. The second block asked about (i) the tasks completed using data analytics tools, (ii) the design activities that could not be done without those tools, and (iii) the number of tools utilized for data processing. The third block asked (i) who else in the same department used the specified tools and (ii) what difficulties the participants encountered using the tools the company made available.

The fifth section of the questionnaire inquired about (i) the potential functionalities and services provided by other tools the designer would like to use and (ii) which tools would not be their first choice. This section comprised three blocks inquiring about (i) the characteristics of tools, (ii) the participants' experience with the tools, and (iii) the expectations for a "true" solution. The first block of questions was intended to learn about (i) the degree of importance and satisfaction with the used data analytics tools, and (ii) how easy they were to master. The second block collected information about (i) the participants' most liked and disliked means and (ii) the tool functionalities the participants knew of but did not use. The questions in the third block placed the participants in a situation where they could (i) assume an instrumentation that would assist designers in other design activities by executing many additional data analytics functions and (ii) conceive a more application-stimulating and designer-friendly data analytics toolbox.

Finally, the last section of the questionnaire collected extra information or suggestions that the respondents considered important to communicate and that might enhance the interrogation or the obtained knowledge. In two blocks of questions, the respondents were asked to (i) evaluate the quality of the interrogation in terms of utility, comprehensibility, and time needed and (ii) recommend possible improvements and provide additional information to supplement their answers. The QBI questions are given in Appendix 1.

3.2.6. Interrogation, collection, and organization of raw data

We obtained responses from 83 participants, but only 40 completed the entire questionnaire. Based on the obtained background information, only 27 participants were chosen for further analysis from this group of subjects. This filtering was done based on the first question, which asked participants (predominantly product designers) whether they were engaged in designing white goods (the target products of the study). The cardinality of the sample (27 full-fledged participants) was greater than the minimum sample size discussed previously.

To ensure research rigor and to increase the validity of outcomes, we assigned an impact value to the participants' answers (to weight the significance of the participants). The assigned impact value varied between zero (no impact) and 10 (very strong impact). These values were assigned to all answers obtained for the first two sections of the questionnaire. We did not assign impact values to questions if they were informative in nature or if they offered the chance for the respondents to present their own ideas or to explain their choice. The weights of the participants played an important role in determining the influential profiles for the study. For example, a need revealed by five participants with high weights was given higher priority in the follow-up phase of our research than a need revealed by seven participants with low impacts. Below, we present statistical information about the participants based on their answers to the first two sections of the questionnaire.

All of the 27 participants were highly educated – 70% have a Ph.D., and 30% have a master's degree – based on which we assigned the same impact value (I) to all of them. Participants' years of experience in design broke down and were weighted as follows:

- 4% had more than 40 years (I = 8)
- 30% had between 20 and 40 years (I = 9)
- 7% had between 10 and 20 years (I = 10)
- 19% had between 6 and 10 years (I = 10)
- 11% had between 3 and 6 years (I = 7)
- 7% had between 1 and 3 years (I = 6)
- 22% had less than 1 year of experience (I = 4)

Regarding participants' years spent at their current institution:

- 44% had between 20 and 40 years (I = 10)
- 7% had between 10 and 20 years (I = 10)
- 19% had between 6 and 10 years (I = 10)
- 19% had between 3 and 6 years (I = 9)
- 7% had between 1 and 3 years (I = 8)
- 4% had less than 1 year (I = 4)

Regarding participants' years spent in their current job or position:

- 30% had between 20 and 40 years (I = 10)
- 15% had between 10 and 20 years (I = 10)
- 4% had between 6 and 10 years (I = 10)
- 22% had between 3 and 6 years (I = 9)
- 22% had between 1 and 3 years (I = 9)
- 7% had less than 1 year (I = 4)

Regarding the designation of the participant’s job or position:

- 18% were development process managers (I = 9)
- 15% were senior product designers (I = 10)
- 11% were researchers (I = 5)
- 7% were project managers (I = 6)
- 7% were strategic product designers (I = 10)
- 7% were research group coordinators (I = 5)
- 4% were young detail designers (I = 9)
- 4% were product managers (I = 7)
- 4% were heads of a manufacturing technology section (I = 5)
- 4% were principal scientists (I = 4)
- 4% were digitalization managers (I = 8)
- 4% were software engineers (I = 6)
- 4% were executive directors of research, brand, and strategy (I = 5)
- 4% were project leads of innovation networks (I = 8)
- 4% were engineers (I = 5)

As for participants’ usage of product-data-induced information in their daily tasks, one participant did not answer the question (I=0). The distribution and weighting of the remaining participants was as follows:

- 52% used data regularly (I = 10)
- 26% seldom used it (I = 6)
- 7% used it exclusively (I = 5)
- 11% of them did not use it at all (I = 10)

Since innovation and use of recent technologies are as important in some countries as in others, we introduced another indicator based on the country where the participant’s company is located. We obtained the following distribution:

- 63% were from developed countries (I = 10)

- 11% were from developing countries (I = 7)
- 22% were from under-developed countries (I = 5)
- 4% did not answer this question (I = 0)

Having the personal profiles of the participants and information about their general work environment, we processed the data concerning the products they designed and the targets, focus, innovation, and strategy of their companies. In terms of the type of white goods the participants designed, we assigned an impact value of I = 10 to all participants, since this question was an informative one and they all were confronted directly or indirectly with white goods.

For participants' answers on the types of innovation their company supported, we assigned an impact value of I = 1 to basic research, I = 6 to breakthrough innovation, I = 7 to disruptive innovation, I = 8 to incremental innovation, I = 8 to sustaining innovation, and I = 10 to strategic innovation. Many participants chose more than one option. In these cases, we calculated the overall impact value by averaging the individual impact values of the chosen options and rounding up this average. We obtained the distribution shown in Table 3.2.

Concerning the business strategy of a participant's company, we assigned an impact value of I = 10 to leadership, I = 9 to differentiation, I = 7 to focus, I = 6 to cost leadership, and I = 5 to research and sustainable design. Again, some participants chose more than one option. In such cases, we calculated the participant's impact value as described in the previous paragraph. After calculations, we obtained the distribution shown in Table 3.3.

In the course of processing the responses, we used the so-called "first transition question" to decide on who would take part in the rest of the statistical evaluation. This transition question asked the participants whether they used big data in their daily design tasks. The question allowed them to choose from among the options "yes," "no," "not relevant," and "not known." Those who chose "no" or "not known" were not accounted for in processing the answers related to this section, which inquired about the use of big data and asked participants to characterize their use of big data in their design activities. The following distribution of answers to this transition question was obtained:

- 56% answered yes (I = 10)
- 33% answered no (I = 0)
- 7% answered not relevant (I = 1)

Table 3.2. Findings concerning types of innovation

I	10	9	8	7	6	5	1
%	4	4	19	33	15	7	18

Table 3.3. Findings concerning company strategies

I	10	9	8	7	6	5
%	18	52	4	11	4	11

- 4% answered not known (I = 0)

For answers to the questions about the stages of the product lifecycle from which data are collected for improving product features and design, we assigned the following impact values and obtained the distribution figures:

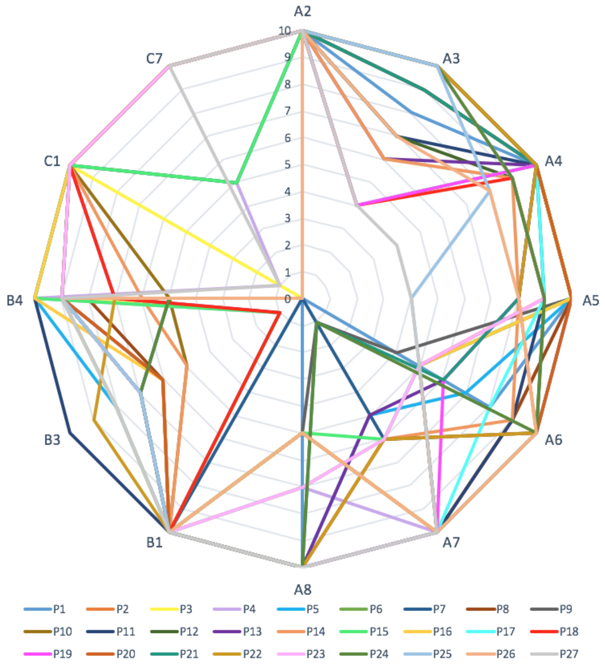
- 37% collected data from all stages (I = 10)
- 11% collected data from the BoL stage (I = 5)
- 11% collected data from the MoL stage (I = 10)
- 4% collected data from the EoL stage (I = 5)

For the sake of completeness, it must be mentioned that 37% of the participants did not answer this question (I = 0).

In Figure 3.2, we graphically represent the impact values assigned to participants' answers. Note that this representation was used to organize the research data and to filter out the participants who would not be taking part in the further analysis. Based on the assigned impact values, we averaged the significance of the opinions of each participant and expressed it as a weight using the following formula:

$$W_{pj} = \sum \frac{I_{Ai}}{N}, \tag{3.2}$$

where W_p is the weight of the significance of the participant, j is the number of



where:

- *A_x are questions from the 1st section of the interrogation*
- *B_x are questions from the 2nd section of the interrogation*
- *C_x are questions from the 3rd section of the interrogation*
- *P_x are participants*

Figure 3.2. Impacts of participants' answers

participants, I_{Ai} is the impact of the answer to question A_i, i is the sequence number of the questions, and N is the total number of questions with an allocated impact. Having calculated the opinion significance weight for each participant, we found that all weights were between 5 and 9. This meant that all participants were appropriate to the purpose of the interrogation and it was correct to consider them in evaluating the responses.

For the open-ended questions, we used coding to detect similarities and differences between the answers provided by participants. The objective was to cluster answers with similar semantic content. We identified 22 clusters within the entire set of open questions. Every cluster represented a generalized need of product designers concerning data analytics tools, as shown in Table 3.4.

3.2.7. Processing the research data and outcomes

To identify the needs of designers in terms of data analytics tools, we included a second “transition question” in the fourth section of the questionnaire. This question asked participants whether they were using data analytics software tools in their daily product development tasks. Altogether, 21 out of the 27 contributing participants answered “yes” to this question. We considered the responses of these 21 participants in evaluating the rest of the questionnaire. The first questions in the fourth section were informative in nature. They concerned (i) the tools designers were familiar with, (ii) the tasks they executed using them, and (iii) who else within their company also used these tools. The remaining questions in the fourth section were directly or indirectly oriented towards the designers’ actual needs. These pieces of information were strategically important for our research: based on them we could build a knowledge platform for our follow-up research that explains what developers and designers of white goods miss in the existing data analytics tools and what they would expect from the next generation of such tools.

In response to a question about the difficulties they have encountered using their current tools, 71% of respondents mentioned that they faced one or multiple problems:

Table 3.4. Explored clusters of responses to open questions

Clusters			
Code	**Name**	**Code**	**Name**
C_1	Multiplicity of functionalities	C_2	User assisting
C_3	Ease of use	C_4	Performance in terms of data types
C_5	Adaptability	C_6	Variety in visualization
C_7	Affordance	C_8	Computational power
C_9	Efficiency	C_{10}	Flexibility
C_{11}	Smartness	C_{12}	Connected
C_{13}	Configurable interface	C_{14}	Portability
C_{15}	Data collection	C_{16}	Usability
C_{17}	Technical requirement	C_{18}	Highly specialized
C_{19}	New releases	C_{20}	Freedom of use
C_{21}	Large set of tools	C_{22}	Integrated framework

Table 3.5. Most disliked features about data analytics tools

Cluster code	%	Statement
C_1	5	Lack of needed functionalities to analyze data
C_2	24	Bad learning curve and customer support, slow learning time, the absence of good training, learning program writing, and instruction information
C_3	9	Lack of ease of use caused by the heterogeneous user interfaces and the complexity of programming
C_4	14	Combination of qualitative and quantitative data is still a challenge, low performance for big databases, and the disliked proprietary format extension for saving files
C_5	9	Lack of adaptability to different design tasks and the complexity of interpreting the outcomes of the tools
C_6	9	Lack of desired variety of visualizations and inadequacies of data display
C_7	9	High cost of software tools and packages
C_8	5	Unresolved bugs of the tools
C_{11}	5	Nonintuitive
C_{15}	9	Dissatisfaction with the transferability of output data among tools and the enormous amount of time involved in collecting relevant data
C_{19}	5	Getting confused by new releases of tools

- 33% of designers cited the complexity of the user interface
- 33% referred to the complexity of programming
- 38% mentioned the complexity of data processing
- 29% of them faced no difficulties

When we asked respondents about the things they most disliked about the current tools, 71% had an opinion to express. We used the already introduced clusters to categorize these answers. The answers are represented in Table 3.5, where CC stands for the code of the clusters of need, % is the percentage of participants who responded, and the "statement" column summarizes what participants said.

As a follow-up to this question, we asked participants to describe a data analytics tool in terms of the functionalities or computer support that would assist them in their design

Table 3.6. Designers' expectations regarding new data analytics computer support

Cluster code	%	Statement
C_1	33	Complete tool with high performance
C_2	19	Assisting user step by step
C_3	19	Advising users in their choices
C_4	24	Combining data from multiple sources
C_5	19	Providing semantic support for data
C_6	5	Proposing multiple visualization options
C_7	5	Affordable to get
C_{10}	5	Flexible in terms of allowed tasks
C_{11}	5	Intuitive tool (smart)
C_{12}	9	Everywhere connected
C_{13}	9	Customizability of the tool
C_{14}	19	Accessibility of the tool at any time

activities, based on their objectives and tasks. The answers were arranged according to the specified clusters of needs. Table 3.6 summarizes the responses. The order in Table 3.6 was determined by the percentage of participants who identified a specific need pertinent to the concerned cluster.

3.2.8. Summary of findings

After weighting, classifying, and ordering the answers, we concluded that the hypothesis made in the beginning of the study concerning an effective methodological approach was correct. The established was true. Based on the specified research model and research design, we obtained valuable insights into what was missed by the interrogated designers and what their concrete expectations for data analytics tools were in the context of supporting product improvement. Consequently, the objectives of the empirical part of the investigation (the questionnaire-based study) were also achieved.

Considering the findings (including the established clusters of needs) presented in Tables 3.5 and 3.6, we can provide a generic formulation of the needs (Ns) of white goods designers. We have captured these needs in the following expressive phrases:

- N_1: Step-by-step assistance
- N_2: ***Advice at means of selection***
- N_3: Multifold data visualizations
- N_4: Multichannel data management
- N_5: ***Blending of datasets***
- N_6: Combining qualitative and quantitative data
- N_7: Permanent accessibility
- N_8: ***Adaptation to user***
- N_9: Case-based reasoning

- N_{10}: Learning from applications

The first need was rooted in the complexity of the functioning and use of data analytics tools. Designers need assistance at every step and context throughout the entire design process. The second need was formulated based on the desire for an effective application of the DATs of dynamically changing function and the desire to concentrate on product enhancement issues rather than on the exploration and exploitation of the computational functions. This issue can be addressed if the tool contains a product-, process-, objective-, and context-sensitive advisory mechanism that can be tailored to the application environment. In this way, the challenge posed by an overwhelming set of tools in one package or toolbox can be overcome, and information about pertinent tools on the market can be provided. Should these two needs be fulfilled, the tool selection time, the learning time, and the effort required to use the tool will be reduced and the cognitive overload of designers lessened.

The third need reflects the dissatisfaction of designers with the currently available data visualization and display choices and expresses their desire to have multiple smart data visualization options that capture semantics and can be varied based on the executed procedure, the nature of data inputs, and the expected outputs. A solution to this issue is offering smart visualization mechanisms instead of multiple tools. Dedicated techniques for dynamic, product-behavior-dependent visualization of MoLD associated with the working product seems to be a proper direction for tool development. These can be used not only in monitoring product operation but also in supporting product maintenance. Eventually, this can feed decision-making about product improvements.

The fourth, fifth, and sixth needs are triggered by the variety of data sources and data types and the lack of semantic incongruences. The phrases $N_4 – N_6$ indicate the sources of problems and give some initial clues for solving them. For instance, the difficulties encountered by designers in exporting and combining different types of data can be traced back to (i) version mismatches of software tools, (ii) the lack of methods for autonomous combining of data from multiple databases with different external schemas, and (iii) the incompatibility of data streams from multiple data sources (e.g. heterogeneous sensor networks). Current tools perform poorly in integrating big datasets, not to mention the issue of semantic fusion of heterogeneous datasets. These needs indicate the need for intensive research in this area as well as the demand for novel computational methodologies.

A recognized challenge is the combination of qualitative and quantitative data (e.g. image information, sounds and speech, and text with structured numerical data beyond annotation). This also concerns MoL datasets, which are known to be huge in volume and highly diverse and, more importantly, can be obtained in multiple ways. They can be aggregated from field observations, interrogations of users, failure log files, maintenance reports, or relevant web resources such as social media and user forums. Alternatively, they can be elicited directly by sensors in or self-registrations by smart products. Comprehensive management of MoLD requires the combination of all pertinent datasets (usage, sensors, maintenance, and servicing data) before analysis. In turn, the outputs of such comprehensive processing can provide valuable inputs for enhancement of products and services. In addition, the problem with the transferability of the output data and the time spent in collecting the relevant data should be addressed.

The seventh need expresses the intent of designers to possess tools that are easily accessible independent of the location or the circumstances of working. A potential solution can be (i) everywhere connected resources, (ii) ubiquitous reachability, (iii) software-as-a-service, or (iv) any combination of these. A technical solution is (i) using cloud environment and semantic web technologies that support parallel execution, (ii) capturing of the meaning of data, and (iii) incorporation of web-services into the data analytics workflow. By means of these, the DATs become pervasive, anywhere and anytime accessible, and the datasets will be enriched.

The eighth need points to next generation customizable and reconfigurable tools. Today, tailoring to the demand of designers is based on creating personal accounts and user profiles. In the future, the tools can learn the user and build behavioral models. Having recognized the user, the tool will also shape itself to the project and the type of product and will reason about support and innovation opportunities. This necessitates dealing with new privacy and dependability challenges, but it also allows designers to add or delete options according to their concrete needs.

The ninth and the tenth needs express the desire for more intuitive and smarter tools. Case-based reasoning (CBR) and various unsupervised learning algorithms should be intensively used towards this end. CBR is an artificial intelligence technology that facilitates solving new problems based on the solutions of previous similar ones [6]. It can be used on its own or in combination with other learning modalities such as neural network-based or probability-based learning [7]. Obviously, solutions from past designs cannot always be used directly in new cases. Learning and abstracting from many different applications can contribute to the generation of better-fitting solutions to new problems.

Smart tools are supposed to be able to make the necessary adaptations of past designs in a quasi-automated manner. Self-adaptation and self-evolution of products should also be considered as future capabilities [8]. The dual (tool and product) self-adaptation offers new opportunities never experienced previously. Clearly, achieving this requires transdisciplinary research efforts and high-level synergy among data analytics, artificial intelligence, engineering design, and human/social factors. Should this happen, smart tools will be able to provide semantic support to product redesign and innovation based on reasoning with fused data streams and patterns. The patterns may be extracted from MoLD according to the type of improvements the designers want to bring to the product.

3.2.9. Setup and conduct of the comparative study

Although all explored needs are concrete and rooted in current practice, their solution lies the near future. In this context, two questions arise: (i) Are the identified needs realistic in a broader context? and (ii) Is the fulfillment of these needs supported by research knowledge, evolving technologies, and system engineering strategies? This calls for validation of our findings in this broader context. To this end, we conducted a literature study to triangulate the results of the interrogative study with the published results in the above three aspects. We present this comparison of the needs extracted from participants' answers to the questionnaire and the propositions released for public debate in the literature. The ultimate objective is to determine to what extent the practice and the theory are in overall harmony rather than to identify particular differences in the formulated needs.

As a first step, the phrases used to describe the abstracted needs were reused to develop a set of keywords as search terms for finding relevant academic and professional publications. In conducting our keyword-based Internet search, we observed a lack of articles and papers heading in the same direction. There were a number of papers on pertinent topics, but they were not tailored to our particular research domain. Researchers have addressed many challenges associated with BDP, including aspects such as (i) interaction and user interface, (ii) visualization of large amounts of data, (iii) databases and storage, (iv) processing algorithms, (v) network infrastructure for data transfer and transport, (vi) uncertainty qualification, (vii) parallelism and duplication reduction, and (viii) domain libraries, development libraries, frameworks, and tools [9]. These topics, however, were not addressed in the context of enhancing white goods.

In comparing the needs identified in the interrogative study with the outcomes of the Internet-based literature search, we found several interactive guides concerning N_1 and N_2. For instance, software components were proposed for advising users on selecting tasks [10], but no information was given regarding the steps to follow in applying data analytics tools in the product enhancement context. With regard to need N_3, despite the constant efforts to improve visualization options, multi-aspect dynamic visualization of large quantities of data and information is still limited [11]. The importance of this issue makes it a key area for improvement [12]. Many authors, claiming that this phenomenon remains a challenge, emphasized the importance of needs N_4 and N_5.

The complexity of the processed dataset is linearly proportional to the level of complication to manage combining multiple datasets [13]. Combining multiple data sources is valuable for creating knowledge, but implementation of the data integration and fusion is difficult [14]. Reasons for this are both the unavailability of dedicated tools and the under-development of existing software tools, which do not allow even the integration of multiple data sources and formats [15]. Concerning N_6, many software tools allow data to be combined within one dataset, but combining datasets of different structures and natures is still in its infancy. Practical techniques for semantically combining qualitative and quantitative data also need further studies and computational solutions [16].

As far as need N_7 is concerned, most currently available big data analytics tools are workstation oriented, and only a minority are offered by the major cloud service providers (who make them accessible at any time) [17]. As for need N_8, it was not discussed at all in the context of DATs in the related papers. It seems that the issues of CBR and learning from applications, conveyed by needs N_9 and N_{10}, cannot be separated in the mirror of the current trend of research. These issues appear in combination in many publications [18] and are addressed in different application contexts. To operationalize these functionalities, data analytics tools should include algorithms that can access and process database and warehouse contents. The use of similarity evaluation procedures is proposed to retrieve the most similar design solutions from these sources [19]. However, we could not find papers about the extent to which this approach can support non-incremental product innovation or disruptive enhancement.

3.2.10. Conclusions

The two completed strands of our research explored the needs associated with the daily data analytics and product enhancement practices of white goods developers and the potential for fulfilling the formulated needs based on current knowledge and massive data management methodologies and technologies. Putting together these two aspects allowed us both to reflect on the reality and feasibility of the needs and to associate the needs with possible new functionalities and services of next generation data analytics tools. Our understanding is that most of the formulated needs point to the necessity of increasing the smartness of analytics tools and environments in terms of both data processing and human–system interoperation. It can be anticipated that future developments should address not only interfacing issues but also issues related to interoperation of humans and system actors.

Both the interrogative study and the web search-based study indicated that product designers, in particular white goods developers, seldom use MoLD to enhance the function and implementation of their products and that the repertoire of data analytics tools used is usually limited. White goods designers typically do not or cannot use their DATs to convert product data into problem-solving knowledge that could serve as the basis for idea generation for both incremental product enhancement and the creation of a new generation of products.

Our main conclusion is that the time has come for tailoring data analytics tools and environments based on the specific needs and operational contacts of product designers. One proposition is that it should include the development of smart (context-sensitive and context-adaptive) mechanisms that provide more insightful data management approaches for designers. The currently applied tools are (i) functionally complex, (ii) programming intensive, and (iii) require the application of a variety of skills [20]. As a next proposition, we suggest that the expected data analytics solutions should be in harmony with the multiplicity and heterogeneity of data collection practices and analytical needs and should be able to cope with incomplete data [21].

Furthermore, if we want to meet the identified needs with data analytics toolbox functionalities and services, this must happen in a systematic manner by using the proposed or similar clustering. Our proposition is that the expected functions should be divided into three major categories:

- Novel interface functions that can be preprogrammed and realized without any modifications to commercial tools. These can be a solution to needs N_1, N_2, N_3, N_7, and N_8.
- Sophisticated data management functions that can be implemented as auxiliary functions of new toolboxes, to allow merging multiple data streams, facilitating data fusion, increasing computational performance, improving usability, and facilitating human interpretation. They can be solutions to needs N_4, N_5, and N_6.
- Smart semantic and procedural reasoning functions that use system intellect provided by artificial intelligence and system learning mechanisms, context information processing, situation awareness, strategy developments, and system adaptation capabilities. Ultimately, these are expected to support addressing all the

needs and to allow the extraction of meaning from MoLD and from other lifecycle data.

Our last proposal is that - having these novel affordances - developers and designers of white goods can optimally benefit from processing product, process, and context data, and can generate innovative ideas for improvement of current products and for the creation of brand new products.

3.3. Exploring the opportunity for using axiomatic theory fusion

3.3.1. Fundamental notions and specific objective

A theory is defined as “a set of interrelated assumptions, concepts, and definitions that presents a systematic view of phenomena by specifying relationships among variables, with the purpose of explaining and predicting the phenomena” [22]. Every theory is recognizable and reducible in terms of its logics, constructs, models, and mechanisms. The majority of theories are monodisciplinary or interdisciplinary. From the perspective of transdisciplinary system development, such theories sometimes create constraints and limitations. Limitations may also be encountered in certain domains of interest due to the lack of comprehensive, descriptive, explanatory, and/or predictive theories. It would be obvious to enhance the descriptive, explanatory, and predictive power of theories by integrating the related potentials of more than one theory. Integration of theories would also reduce the need for development of theories by explorative research. This is now an important issue, since in some research areas there is little current literature on methodologies for combining existing theories. Nonetheless, it is worth considering knowledge synthesis based on systematic integration of tested theories for the above two reasons. The expected new opportunities stimulated our research in this direction.

Existing engineering design theories are not well-defined [23] and are insufficient to provide a systemic design method [24]. Thus, combining or integrating theories is also required in these fields of application, namely software tool engineering, complex system engineering, and product and service design. In these domains, not only quantitative theories but also qualitative ones play important roles. While the need for integrating qualitative theories prevails, combining them with other qualitative and quantitative theories is challenging, due to the concomitant need for semantic interpretation and logical merging with quantitative chunks of knowledge. First, the issue of interpretation is raised by the differences in definitions, interpretations, vocabularies, terminologies, and context dependencies between theories. Second, it is entailed by the complexity and heterogeneity of theories that should be considered in the transdisciplinary designing of artifacts and services.

The specific domain of our research is conceptualization, implementation, and validation of new-generation SDATBs. These tools have reached a level of sophistication that often challenges nonspecialist users, such as product designers [25]. Other challenges emerge from real-time generation of big data, the multiplicity of data sources, and the nature of products [26]. There is no specific theory in the literature that would cover all important aspects of the process of developing a next generation SDATB. On the other hand, as has been hypothesized, conceptualization, implementation, and validation of an SDATB may be efficiently supported by synthesized theories that include technological, human, social, computational, and organizational chunks of knowledge. In principle, the underpinning theory could be generated by exploiting the different epistemological relationships between theories. Namely, if we suppose that *N* partially interoperating qualitative theories exist, they may be in various relationships with each other as shown in Figure 3.3. For the purpose of this study, we considered bridging [27], complementary [28], overlapping [29], extending [30], or refining relationships [31].

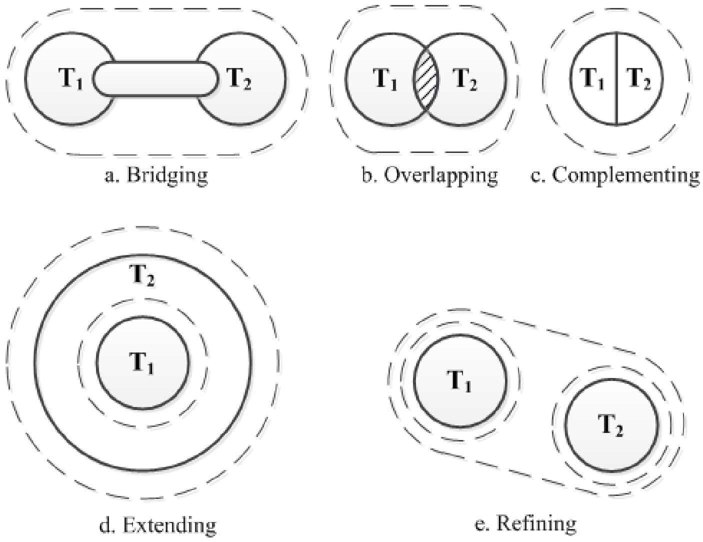


Figure 3.3. Possible epistemological relationships between theories

Our concrete objective is to start out with *N* theories that are relevant from the perspective of smart data analytics tools development and synthesize from them a theory more powerful and comprehensive than its independent component theories in terms of description, explanation, and/or prediction. Unfortunately, the literature does not provide a solution and does not advise on how to get to such a result methodologically. Having a correct methodological solution would provide multiple advantages, such as sparing us the need for extensive new research and saving time for multiple industrial practices. There are several theoretical foundations and frameworks that may be used as means, but their exploitation lags, as is shown by the related literature.

3.3.2. State of the art of generic theorizing approaches

Scientific theories are the basis for the description, explanation, and prediction of how and why phenomena occur [32]. They are generated by scientific research under dependable conditions and as the fundaments of technology [33]. Two types of scientific theories are typically distinguished: (i) quantitative theories, which are based on observable and measurable facts and proper (true in a given context) logical relations, and (ii) qualitative theories, which are typically based on observable nonquantifiable facts that can be captured qualitatively [34]. Various logical and methodological approaches have been proposed to support the practice of deriving quantitative and qualitative theories. While theories were traditionally derived in (i) retrospective [35], (ii) inductive [36], (iii) deductive [37], and (iv) abductive manners [38], current data-driven science attempted to formulate theories by extracting patterns from massive datasets and converting the coherent patterns into a knowledge framework that enables theories [39].

The entry point of our research was a literature study that collected information about the methodologies that had been proposed for semantic merging of theories and about the procedures that could be applied in our research context. The first observation was that the issue of experimental development of scientific theories was broadly addressed in the literature, butsuch development remained an exhaustive and complex process [40]. Similarly, development of synthetic theories on the basis of rules and axioms [41] received insufficient attention [42]. Suddaby et al. identified primary, secondary, and tertiary approaches to theorizing: (i) the logic of pure research (emphasizing the enduring structural content of scientific theory), (ii) the logic of induction (emphasizing the interpretation of patterns inherent in empirical data), (iii) the logic of problem-solving (emphasizing practical action and an open and interdisciplinary community of experts), (iv) strong-paradigm logic (emphasizing the articulation of procedures to solve outstanding puzzles within paradigmatic communities), and (v) the logic of emancipation (emphasizing subversive challenges to prevailing knowledge assumptions) [43]. The different theorizing approaches presented in [43] are summarized in Table 3.7.

Below, more insights are given concerning the state of the art of theory development and approaches to systematically merging theories. In particular, we focus on the issue of semantic fusion of qualitative theories. Designers deal with complex phenomena that cannot be covered by simple individual theories [44]. The complexity comes from the multidisciplinary nature of design problems [45]. For this reason, multidisciplinary theories, or a merging of theories from different domains, were needed to cover these issues [46]. Unfortunately, the literature lacked frameworks and methods to merge theories. This was caused by the complexity of the procedures implying a change of constructs and relationships of original concepts of individual theories [47].

Primarily, the existing approaches to merging individual theories aimed at bridging the gaps between specific disciplines [48]. These approaches ranged from the selective borrowing or incorporation of elements of one theory within another, dominant theory to full-fledged blending of method theories in an attempt to generate "new" theories [49]. In this context, concerns were increasingly raised in the literature about the

Table 3.7. Map of different theorizing approaches [43]

	Theorizing within one body of literature	Theorizing across multiple bodies of literature
Theorizing with implicit assumptions of the literature	Problematizing	Combining epistemologies Metaphoric bricolage
Theorizing with explicit constructs of the literature	Contrasting Practical rationality Inductive top-down theorizing	Blending

tendency to combine theories with different or even incompatible underpinning ontologies and epistemologies, as well as about the challenges that this tendency represents when reconciling conflicting assumptions in the process of theory development [50]. Such concerns also emerged in interdisciplinary accounting research [51], as well as in the broader management and organization literature [52]. They warranted serious considerations regardless of how extensive the blending of the concerned theories was [53].

Systematic combining is an approach for “handling the interrelated elements in the research work” that occurred because “the intertwined activities in the research process” required the researcher to “constantly [go] back and forth from one type of research activity to another and between empirical observations and theory” [54]. Rough set theory merging reuses the knowledge that was already contained in available repositories of computer-checked mathematical knowledge or that could be obtained in a relatively easy way [55]. It shows interconnections to enumerate between fuzzy sets, lattice theory, topology, and category theory, among many others. It also makes clear that formalization of the topic can be a challenging task. This task can result in the discovery of new connections, especially after the application of machine learning tools [48]. Another method of merging concerns the ontological modeling approach to blending theories used for instructional and learning design [56]. This approach has been used to model educational theories. The main concerns include (i) a theory or paradigm-independent ontology for modeling learning and instruction, (ii) compatibility between prescriptive and descriptive models derived from educational theories, and (iii) theory awareness brought out by an ontological modeling framework [57]. The challenge remained in the fusion and integration of theories, since it was far from an easy task to accomplish [58].

3.3.3. State of the art of axiomatic theorizing

An axiomatic system is a logical system that possesses an explicitly stated set of axioms from which theorems can be derived [59]. This approach typically includes deductions from abstract axioms that contain correctly defined concepts related to an empirical event [60]. It is an approach that regards axioms as the basis of the theory, while the other propositions of the theory are obtained as logical consequences of these axioms [61]. The obtained axiomatic theory can be defined as a set of statements of relations among concepts, considering the set of boundary conditions and constraints [62]. The relations within the theory are either captured by axioms or can be derived from them [63]. Axiomatic theories have self-evident premises that can be accepted as true without controversy or much empirical confirmation [64]. They have the nature of consistency, determinacy, and accuracy [65]. Axiomatization has been found a beneficial analytical approach to investigate the validity of projected decision criteria [66].

Axioms are elementary propositions that are so evident they are acceptable intuitively [67]. They may carry a set of undemonstrated propositions, which by contrast, can be vague, accepted intuitively, or established by practice. Testing the properness of an explanation includes determining that an empirical phenomenon is covered by the total set of axioms and postulates [68]. Postulates are also derived from the theory. At times, a postulate is seen as a synonym of an axiom. This is not true, since a postulate is an axiom dealing with a specific matter (true in a specific context) and cannot be seen as a general statement, unlike an axiom [69]. The axioms and postulates are the notional nodes in an argumentative network of the deductively derived target theory [70]. Its descriptive and explanatory power depends on the coverage of the axioms and postulates. There may be relevant or irrelevant nodes from the perspective of the targeted phenomenon [71].

Axioms formulate concepts with explanatory power, symmetry, neutrality, anonymity, and independence [72]. The legacy of the axiomatic approach comes from the fact that other approaches to theory building may not have a strong theoretical underpinning, and it is not always clear which properties they uphold. Related to our background study, our choice is underpinned by the fact that an axiomatic theory can provide a theoretical basis for design processes and help designers in decision-making, reducing complexity and unnecessary repetitions of design procedures and imparting systematization and rationalization to design activities [73].

Axiomatic theorizing is a top-down deductive approach that makes conclusions based on principles and truths. A theory is defined as statements of concepts and their relationships that explain how and why a phenomenon happens [74]. A theory is called deductive when it is gathered in a systematic manner and steps through careful analyses [75]. The very essence of our proposal is a "theoretical arithmetic," in which the theories are the quantities and discretization, relating, merging, deriving, and projecting are the arithmetic operations [76]. These have been operationalized in the methodology of ATF.

Axiomatization specifies the content of a theory in which a set of axioms and postulates are given, and from which a set of propositions is derived [77]. The self-evident nature of axioms and postulates makes them trustable. Referring to our phenomenon, we can conclude that an axiomatic approach forms a base for merging theories, since it (i) is logical, (ii) does not need to be proven, (iii) contains general statements as "axioms," (iv) contains specific statements within specific contexts as "postulates," and (v) contains propositions derived based on general and specific statements (axioms and postulates). This approach is particularly appropriate in the case of design domains, where solving a problem, innovating, or improving requires different domains of expertise. Giving birth to the ATF methodology will cover the lack encountered in some research domains (such as design theories, data analytics theories, and so on), where theories are not up to date regarding processes and new techniques and technologies. Another advantage of this approach is that it allows the development of one's own sufficient theory dedicated to one's particular case and domain of interest.

3.3.4. Basic considerations of the axiomatic theory fusion methodology

As a starting point for the methodology development, the following hypothesis was established: *a robust and comprehensive conceptual basis for a knowledge platform in the context of multidisciplinary research needs to combine many composite theories*.

This was reflected in the conducted literature study. It was found that one existing theory was not enough in some research areas. Integrating theories from different disciplines is becoming a need to develop new relevant theories [78]. The existing theories focused on one aspect separately without integrating all aspects in one theory. Grouping all aspects of research into a new theory was a good strategy for system development.

In the literature, we found different types of frameworks for building theories. They varied based on the domain of interest, the targeted outcome, and so on, but they all agreed that a theory should provide explanations and predictions and should be testable [79]. To elaborate on the constituents of the methodology and their relationships, we shifted the attention in the literature search to principles of theory formation in design. The works in [79] - [80] were summarized and used as the basis for the new methodology framework proposed by Horváth, I., as presented in Figure 3.4.

Following the above scheme of methodology construction, the first step consists of determining the purpose and scope of the research. In this step (i) the problem domain, (ii) its resources, and (iii) the underpinning theory are identified, taking into consideration the problem, the motivation, and the target of the procedure. The second step, called the execution procedure, specifies the process of the theory development including the principles and steps to follow. Then, the pools of methods supporting the realization of the process steps are specified, as are the instruments – also called supporting tools – for applying the methods. Finally, the proper criteria to determine the methodology’s relevance are determined, and the conceived limitations are identified. To this end, the principles implied by the methodology, the knowledge obtained, and the relevance of the procedure are tested. In the upcoming section, all constituents of the ATF methodology are described and discussed.

Colander, D. stated that a pure axiomatic approach attempts to start with a minimal number of assumptions and to arrive at as many conclusions as possible from those assumptions [81]. He also claimed that an axiomatic approach requires parsimony in assumptions. Rigorously following these principles, Horváth, I. proposed a practical approach to theorizing in a deductive manner; this approach is referred to as ATF [82]. The underpinning theory is based on the assumption that a number of properly selected theories can provide sufficient explanations for phenomena that have not been studied experimentally. The multiple theories must be considered simultaneously and must be interwoven in the reasoning. However, these theories are conceptually different and

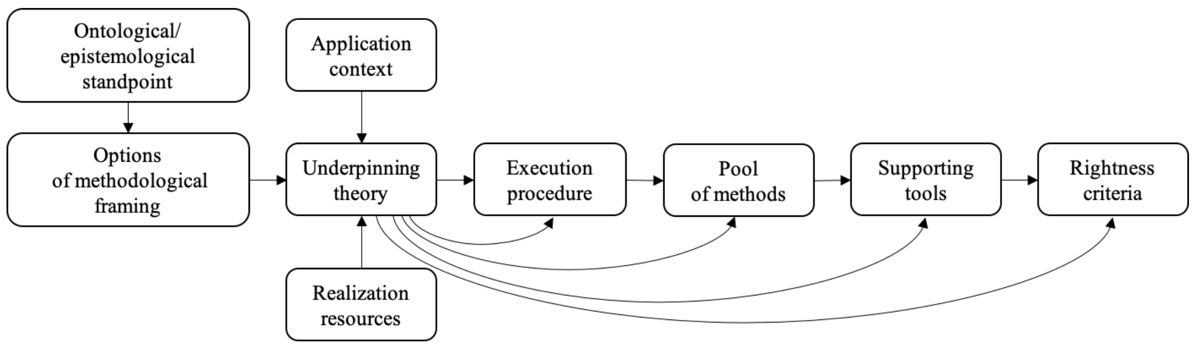


Figure 3.4. Constituents of a generic methodology

cannot be combined straightforwardly. They also may not be completely coherent and consistent. Therefore, a proper new theory should be synthesized by blending the existing in-part-sufficient theories (called source theories) into a new explanatory theory (called the target theory) with sufficient clarifying power and consistency. Figure 3.5 shows the logical steps of ATF, which transforms a set of source theories into one target theory.

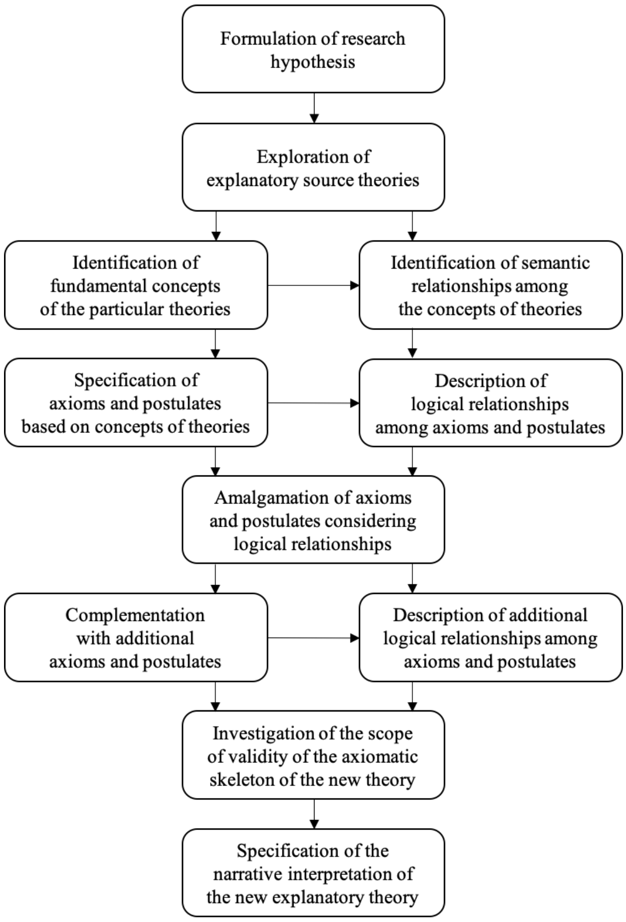


Figure 3.5. The logic of axiomatic theory fusion

The ATF approach was chosen for this purpose. We successfully applied this approach to create a robust knowledge platform [83] coupled with an epistemological description of basic concepts and their relations in a given context. It first explored the context of the study and divided it into subdomains to facilitate investigation of the affordances of theories in the literature that covered each aspect of the studied phenomenon. Here, an identification of possible semantic relationships between aspects was needed. We then chose the appropriate existing theories and identified possible links or relationships between them. The next step was establishing a formal theoretical system using deductive processes, starting with specifying available epistemological entities (axioms and postulates) and defining additional ones to make the epistemological basis of a new theory consistent and complete. It is preferable to add postulates as the only source of subjectivity and keep the original formulation of the theories as axioms. These sets of axioms and postulates for all chosen theories formed the skeleton of the new theory, which we later fused, polished, and converted into the narrative formulation of the target theory.

3.3.5. Process and steps of axiomatic theory fusion

3.3.5.1. Process of axiomatic theory fusion

The process of ATF reflects the logic of deductive reasoning in that it is driven by a proper research hypothesis, and it tries to develop a descriptive, explanatory, or predictive theory by considering relevant existing scientific theories. However, this reflecting is not strong because two issues must be considered. First, a well-founded theory alone might be insufficient to explain the "truth" (properness) of the stated hypothesis due to its possible limited coverage. Second, since a source theory in deductive reasoning is typically broader that the target one, the specialization challenge should also be considered (it is worth noting that an inductive approach would face a generalization challenge concerning its coverage). Therefore, systematic combination of component theories was regarded as a way out of the trap of deduction. Nevertheless, a consistent fusing of multiple component theories needs further considerations. For this reason, the process of ATF consists of five main stages:

- Selection of theories based on their usefulness as source theories.
- Axiomatic discretization of component theories, which is done in two steps: (i) semantic discretization of theories and (ii) arrangement and composition of axioms and postulates structures.
- Semantic and visual capturing of relationships, which is done in three steps: (i) creation of a relationships network, (ii) matrix representation and rearrangement, and (iii) proposition derivation in a given context.
- The actual fusion of the component theories, which is done in three steps: (i) syntactically processing and merging component theories, (ii) deriving propositions based on units of resultant theory, and (iii) transferring propositions into a narrative description.
- Validation of the new theory in the context of the planned application.

3.3.5.2. Selection and semantic discretization of component theories

The first step in ATF was to identify the candidate source theories. To do so, we identified the domains of interest that influence the context of the research. We then transformed them into keywords to establish a literature search to find theories covering every individual aspect. The selected theories were explored for consistency, relevance, and sufficiency. This process determined whether one theory was enough for each domain or an additional theory was needed. Every domain had one global theory that grouped sub-theories if one theory was not enough. Once all chosen theories were selected, we filtered their paragraphs to choose the relevant ones based on their implications for the research context or for the research phenomenon. Paragraphs were judged relevant based on their semantic meaning and their concrete relationship to the application case; paragraphs with similar or redundant meanings were not considered. We then decomposed the chosen paragraphs into a list of short, meaningful statements without modifying their original meaning and without subjective interpretations. This step was called the textual formulation of component theories, which were derived as shown:

$$T_x = \sum_{i=1}^{n} Pr_i \oplus \sum_{j=0}^{m} Pir_j \Rightarrow T_x = \left(\sum_{i=1}^{n'} Sr_i \oplus \sum_{i=n'+1}^{n} Sir_i\right) \oplus \left(\sum_{j=0}^{m} Sir_j\right)$$

$$\Rightarrow T_x^T = \sum_{i=1}^{n\prime} Sr_i, \tag{3.3}$$

where T_x is theory number x, P_r is a relevant paragraph, P_{ir} is an irrelevant paragraph, S_r is a relevant statement, S_{ir} is an irrelevant statement, and $T^T{}_x$ is a textual formulation of theory number x and represents the relevant parts of theory T_x. We call this a textual formulation because we kept the relevant parts as formulated in the theory without subjective transformation.

The operators $\oplus$ and $\sum$ are string concatenation operators. The first one represents the addition of elements and the second one represents the sum of elements next to each other noncommutatively (an example of a noncommutative operation is 2 + 1 = 21). The theory represented in Equation 3.3 is relevant. This means that it contains at least one relevant paragraph. Accordingly, $\sum_{i=1}^{n} Pr_i$ starts from i = 1. The equation reflects that a sequence of relevant paragraphs consists of two parts: a sequence of relevant sentences and a sequence of irrelevant sentences. A relevant paragraph may still contain some irrelevant sentences, but an irrelevant paragraph is just a sequence of irrelevant sentences.

The statements were written as sentences composed of subject entities and relationships between them. For example, if we consider the statement "data analytics generate knowledge," the entities are "data analytics" and "knowledge," and "generate" represents the relationship between them. These entities represented subjects, objects, and nouns, and relationships were the verbs linking them. A statement written in the form *subject + verb + object/noun* was characterized as a "one-to-one" relationship between two entities (subject and object/noun), and a *subject + verb + adjective/adverb* formulation was characterized as a "self-reflexive" relationship (a relationship between an entity and itself) in which the only entity was the *subject*. By considering these rules for all statements, we defined a set of entities for every component theory. This step was called discretization of component theories. The entities were represented using the following notation: $E_{x,i}$, where x is the identifying number of the containing theory, and i is the order of appearance of the entity within the theory T_x. The entities and statements were used to derive both axioms and postulates for all concerned theories.

3.3.5.3. Arrangement and composition of axioms' and postulates' structures

The two previous steps of the methodology were used to determine axioms and postulates from the chosen theories. An axiom is represented as $A_{x,i}$ (where x is the number of the theory, and i is the order of formulation of the axiom). Postulates were distinguished in two forms:

- Postulates derived directly from the theories, represented as $P^D{}_{x,i}$ (where x is the identifying number of the theory, and i is the order of formulation of the derived postulate).

- Auxiliary postulates added based on additional domain or problem knowledge, represented as $P^{A}{}_{x,j}$ (where x is the identifying number of the theory, and j is the order of formulation of the auxiliary postulate).

In this step of processing the theories, we converted all relevant statements into axioms and postulates. This step was called axiomatization of the component theories. Postulates extended the set of relationships. The axioms and postulates of a theory x were formulated as follows:

$$A_{x,i} = (E_{x,j})[R'(E_{x,j}, E_{x,k})](E_{x,k}), \tag{3.4}$$

where R' is an intuitive relationship between the entities derived directly from $T^{T}{}_{x}$, and

$$P_{x,i} = (E_{x,j})[R''(E_{x,j}, E_{x,k})](E_{x,k}), \tag{3.5}$$

where R'' is an intuitive relationship between entities not necessarily derived from $T^{T}{}_{x}$. R' and R'' can be similar in some cases, but they carry different implications. Since a postulate remains an axiom dealing with a specific matter, it is true in a specific context and cannot be seen as a general statement as an axiom can.

The complete set of postulates was represented as follows:

$$\sum_{i=1}^{n} P_{x,i} = \sum_{j=0}^{m} P_{x,j}^{D} \oplus \sum_{k=m+1}^{n} P_{x,k}^{A}. \tag{3.6}$$

Some statement might not be converted directly into axioms and postulates. In these cases, we added additional entities to facilitate the decomposition of the statement. Those entities were added to the first list of entities. Because the original list of entities contains n entities, the numbering of the additional entities should start from $n + 1$. Thus the final list of entities is

$$\sum_{i=1}^{m} E_{x,i} = \sum_{i=1}^{n} E_{x,i} \oplus \sum_{i=n+1}^{m} E_{x,j}. \tag{3.7}$$

The final lists of entities, axioms, and postulates were used as the basis for constructing the relationships network for each theory.

3.3.5.4. Creating relationships networks

Every axiom and postulate captured an elementary statement of the theory in the form of semantic relationships. In this step, so-called relationships networks (RN_x, where x is the identifying number of the theory) were built. The specific objective was to move towards a graphical representation of the complete list of axioms and postulates that could be used as a map of the captured relationships between entities. We first graphically represented the entities and then linked them to each other considering the nature of the relationships between them. These relationships were formulated in the form $R(E_{x,i}, E_{x,j})$, where x is the identifying number of the theory, and i and j indicate the order of appearance of the entities consecutively in the textual formulation of the theory. We distinguished more than one relationship between two entities, since entities could be connected according to more than one semantic interpretation. The established relationships were technically captured as axioms or postulates.

This graphical representation was needed not only to capture the connectivity between entities but also to visualize the unconnected entities and determine the possibility of connecting them through new logical auxiliary postulates to complement the original list of postulates. In this way, an augmentation of the relationships between entities was created. Connectivity between the disconnected parts of the graph was established only if it had a meaningful content and served the purpose. The added set of postulates should not, in any case, be in conflict with the original statements and logic of the source theory.

We graphically represented the relationships network in Microsoft Visio. Figure 3.6 shows an example of the visualization obtained. The circles represent the various entities, the black arrows represent relationships contained in an axiom, the red arrow represents relationships contained in a postulate derived directly from the theory, and the blue arrow represents relationships contained in an auxiliary postulate. The numbers within arrows indicate the numbers of relationships between entities. The relationships network was needed to enrich individual theories, partially contextualize them, and capture the connectivity between entities, but the actual connection could not be detected from it. Thus, another type of representation was needed for this purpose.

3.3.5.5. Matrix representation and rearrangement

A matrix format representation and matrix decomposition were used to expose the connection between all concerned subject entities. Actually, it showed the distribution of relationships among the various subject entities. The matrix M($n \times n$) is a symmetric construct of size n, and the elements of its diagonal are the entities, including their self-reflexive relationships. If there is no relationship between entities (empty slots of the matrix), the value 0 was included for these entities, as shown in Figure 3.7(a). This step was completed using Microsoft Excel.

The matrix decomposition does not represent the semantic distance between entities. For this reason, it was considered as an intermediate step between relationships networks and matrix rearrangement. The rearrangement algorithm reorders the initial matrix into a matrix that brings the entities into a closer (and more expressive) relationship with each other, instead of the initial, largely arbitrary order. In other words, it is a matrix of blocks called Block_M($n \times n$). As such, it is broken down into blocks forming sub-matrixes. This method was used to group the syntactically linked entities. These groups were also subjects of semantic interpretation. The rearrangement resulted in blocks of the matrix, called clusters. Axioms and postulates contained in the clusters were placed next to each other to determine the possible links

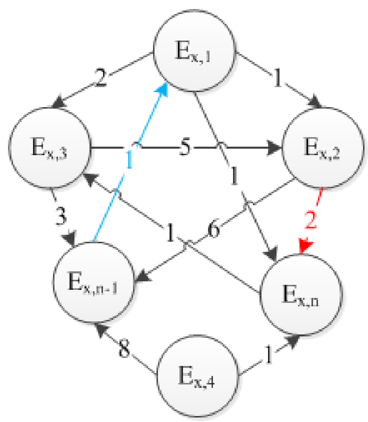


Figure 3.6. Relationships network representation of a theory *x*

	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	$E_{1,4}$	$E_{1,5}$	$E_{1,6}$	$E_{1,7}$	$E_{1,8}$	$E_{1,9}$	$E_{1,10}$
$E_{1,1}$	$A_{1,6}$	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$; $A_{1,5}$; $A_{1,7}$	0	0	0	0	$A_{1,12}$	$A_{1,14}$; $A_{1,15}$
$E_{1,2}$	$A_{1,1}$	$A_{1,13}$	0	0	0	0	0	0	0	0
$E_{1,3}$	$A_{1,2}$	0	0	0	0	0	0	0	0	0
$E_{1,4}$	$A_{1,3}$; $A_{1,5}$; $A_{1,7}$	0	0	$A_{1,4}$	0	0	0	0	0	0
$E_{1,5}$	0	0	0	0	$A_{1,9}$	0	0	0	0	0
$E_{1,6}$	0	0	0	0	0	0	$A_{1,10}$	$A_{1,11}$	0	0
$E_{1,7}$	0	0	0	0	0	$A_{1,10}$	$A_{1,8}$	0	0	0
$E_{1,8}$	0	0	0	0	0	$A_{1,11}$	0	0	0	0
$E_{1,9}$	$A_{1,12}$	0	0	0	0	0	0	0	$P^{D}_{1,1}$; $P^{A}_{1,1}$	0
$E_{1,10}$	$A_{1,14}$; $A_{1,15}$	0	0	0	0	0	0	0	0	0

(a)

→

	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	$E_{1,4}$	$E_{1,5}$	$E_{1,6}$	$E_{1,7}$	$E_{1,8}$	$E_{1,9}$	$E_{1,10}$
$E_{1,1}$	10000	10	11	12	0	0	0	0	13	14
$E_{1,2}$	10	20000	0	0	0	0	0	0	0	0
$E_{1,3}$	11	0	0	0	0	0	0	0	0	0
$E_{1,4}$	12	0	0	40000	0	0	0	0	0	0
$E_{1,5}$	0	0	0	0	50000	0	0	0	0	0
$E_{1,6}$	0	0	0	0	0	0	60	61	0	0
$E_{1,7}$	0	0	0	0	0	60	70000	0	0	0
$E_{1,8}$	0	0	0	0	0	61	0	0	0	0
$E_{1,9}$	13	0	0	0	0	0	0	0	90000	0
$E_{1,10}$	14	0	0	0	0	0	0	0	0	0

(b)

Figure 3.7. The content of the correspondence matrix before and after coding: (a) matrix before coding, (b) matrix after coding

between them. The matrix rearrangement was done using Matlab. Since this software platform deals only with numerical values, a method of coding was introduced to allocate a numerical value or code to the relationships between entities. The coding served purely for identification purposes. It did not indicated any semantic relationship or association of the subject entities or the statements included in the epistemic entities.

Figure 3.8 shows the logic of numerical coding. Two types of codes were applied, one to the diagonal elements of the matrix and the other to the nondiagonal elements of the matrix. To avoid repeating a given code in the case of a rich theory with a large number of relationships, an extension (0000) was added to the entity code of the diagonal elements of the matrix. For this same reason of avoiding duplications in codes, negative codes were also used (e.g. if 69 is the code of the 10^{th} element of entity 6, -60 will be the code of the 11^{th} element of entity 6). A mistakenly repeated code would give false results, since the same code would be applied to different relationships. The final list of codes was used to replace all relationships in the matrix (Figure 3.7(b)).

The matrix rearrangement was done in Matlab using the coded matrix and the following coding:

```
>> T1 = theory1;
   p = symrcm(T1);
   block_T1 = T1(p,p);
```

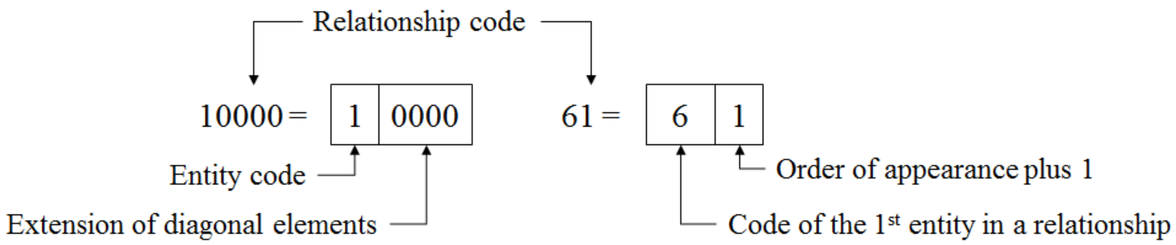


Figure 3.8. Examples of coding

	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	$E_{1,4}$	$E_{1,5}$	$E_{1,6}$	$E_{1,7}$	$E_{1,8}$	$E_{1,9}$	$E_{1,10}$
$E_{1,1}$	10000	10	11	12	0	0	0	0	13	14
$E_{1,2}$	10	20000	0	0	0	0	0	0	0	0
$E_{1,3}$	11	0	0	0	0	0	0	0	0	0
$E_{1,4}$	12	0	0	40000	0	0	0	0	0	0
$E_{1,5}$	0	0	0	0	50000	0	0	0	0	0
$E_{1,6}$	0	0	0	0	0	0	60	61	0	0
$E_{1,7}$	0	0	0	0	0	60	70000	0	0	0
$E_{1,8}$	0	0	0	0	0	61	0	0	0	0
$E_{1,9}$	13	0	0	0	0	0	0	0	90000	0
$E_{1,10}$	14	0	0	0	0	0	0	0	0	0

Coded matrix

(a)

	$E_{1,10}$	$E_{1,9}$	$E_{1,4}$	$E_{1,2}$	$E_{1,1}$	$E_{1,3}$	$E_{1,5}$	$E_{1,8}$	$E_{1,6}$	$E_{1,7}$
$E_{1,10}$	0	0	0	0	14	0	0	0	0	0
$E_{1,9}$	0	90000	0	0	13	0	0	0	0	0
$E_{1,4}$	0	0	40000	0	12	0	0	0	0	0
$E_{1,2}$	0	0	0	20000	10	0	0	0	0	0
$E_{1,1}$	14	13	12	10	10000	11	0	0	0	0
$E_{1,3}$	0	0	0	0	11	0	0	0	0	0
$E_{1,5}$	0	0	0	0	0	0	50000	0	0	0
$E_{1,8}$	0	0	0	0	0	0	0	0	61	0
$E_{1,6}$	0	0	0	0	0	0	0	61	0	60
$E_{1,7}$	0	0	0	0	0	0	0	0	60	70000

Matrix partitionning in Matlab

(b)

Figure 3.9. Matrix rearrangement using Matlab: (a) original content, (b) coded content

Symrcm is the symmetric reverse Cuthill–McKee reordering. It is an algorithm that helps order the rows and columns of a symmetric sparse matrix [84]. In Matlab, the Symrcm transformation function allows having nonzero elements of a matrix closer to the diagonal [85].

Figure 3.9(b) shows the result of the rearrangement of the matrix by Matlab. This rearrangement represented relationships between entities as densified blocks that had not been visible in previous representations.

In the next step, the numerical codes were replaced by the actual relationships within the matrix to identify the axioms and postulates and to simplify the analysis of the ontained blocks. This manipulation was also done in Matlab, using the following code:

```
>> T1 = theory1; % Specification of the theory
   d_num = X; % Specifying X, where X is included in the range of the codes
   [I,J] = find(A==d_num); % find X
   L_I = length(I);
   C=num2cell(A);
```

This action was repeated for all numerical codes of the matrix. Figure 3.10(b) shows the final matrix in Matlab with all relationships explicitly indicated. The next step was to

	$E_{1,10}$	$E_{1,9}$	$E_{1,4}$	$E_{1,2}$	$E_{1,1}$	$E_{1,3}$	$E_{1,5}$	$E_{1,8}$	$E_{1,6}$	$E_{1,7}$
$E_{1,10}$	0	0	0	0	14	0	0	0	0	0
$E_{1,9}$	0	90000	0	0	13	0	0	0	0	0
$E_{1,4}$	0	0	40000	0	12	0	0	0	0	0
$E_{1,2}$	0	0	0	20000	10	0	0	0	0	0
$E_{1,1}$	14	13	12	10	10000	11	0	0	0	0
$E_{1,3}$	0	0	0	0	11	0	0	0	0	0
$E_{1,5}$	0	0	0	0	0	0	50000	0	0	0
$E_{1,8}$	0	0	0	0	0	0	0	0	61	0
$E_{1,6}$	0	0	0	0	0	0	0	61	0	60
$E_{1,7}$	0	0	0	0	0	0	0	0	60	70000

Matrix partitioning in Matlab

(a)

	$E_{1,10}$	$E_{1,9}$	$E_{1,4}$	$E_{1,2}$	$E_{1,1}$	$E_{1,3}$	$E_{1,5}$	$E_{1,8}$	$E_{1,6}$	$E_{1,7}$
$E_{1,10}$	0	0	0	0	A1,14 ; A1,15	0	0	0	0	0
$E_{1,9}$	0	PD1,1 ; PA1,1	0	0	A1,12	0	0	0	0	0
$E_{1,4}$	0	0	A1,4	0	A1,3 ; A1,5 ;	0	0	0	0	0
$E_{1,2}$	0	0	0	A1,13	A1,1	0	0	0	0	0
$E_{1,1}$	A1,14 ; A1,15	A1,12	A1,3 ; A1,5 ;	A1,1	A1,6	A1,2	0	0	0	0
$E_{1,3}$	0	0	0	0	A1,2	0	0	0	0	0
$E_{1,5}$	0	0	0	0	0	0	A1,9	0	0	0
$E_{1,8}$	0	0	0	0	0	0	0	0	A1,11	0
$E_{1,6}$	0	0	0	0	0	0	0	A1,11	0	A1,
$E_{1,7}$	0	0	0	0	0	0	0	0	A1,10	A1,

Matrix coding in Matlab

(b)

Figure 3.10. Matrix coding in Matlab: (a) original content, (b) codes replaced by relationships

study the relevance of the clusters reflected by the blocks in the matrix. We found that some of the clusters were irrelevant, since the axioms and postulates represented by them were too general, or they were conceptually far from the specific application context. In line with this, in the steps below we treat only those blocks that were judged to be relevant or at least partially relevant.

3.3.5.6. Deriving propositions in a given context

The objective of the previous steps was to derive meaningful propositions based on the set of axioms and postulates. We achieved this by analyzing the meaningful blocks of the matrix, which we called clusters and which are presented in the rest of the dissertation in the form $C_{x,j}$, where x is the identification number of the theory, and j is the sequential order of the cluster in the matrix. The clusters were studied separately, and the relationships in each were semantically grouped (whenever possible) and put into the context of the study, or they were ignored. We observed that this contextualization brought the disconnected (or partially connected) elements together. In this stage, the component theories were semantically and contextually integrated in the application context. This procedure provided a first hint of the consistency of the target theory, and it showed whether the chosen theories were sufficient to produce knowledge or additional theories should be investigated.

Propositions ($Pp_{x,i}$, where x is the identification number of the theory, and i is the order of writing of the proposition) represent a logical combination of axioms and postulates contained in the same cluster (they can be many) with regards to the application context. Figure 3.11 shows the steps from the extraction of entities to the determination of propositions for one theory. The propositions were categorized as (i) relevant, (ii) partially relevant, or (iii) irrelevant based on their implications and their importance in

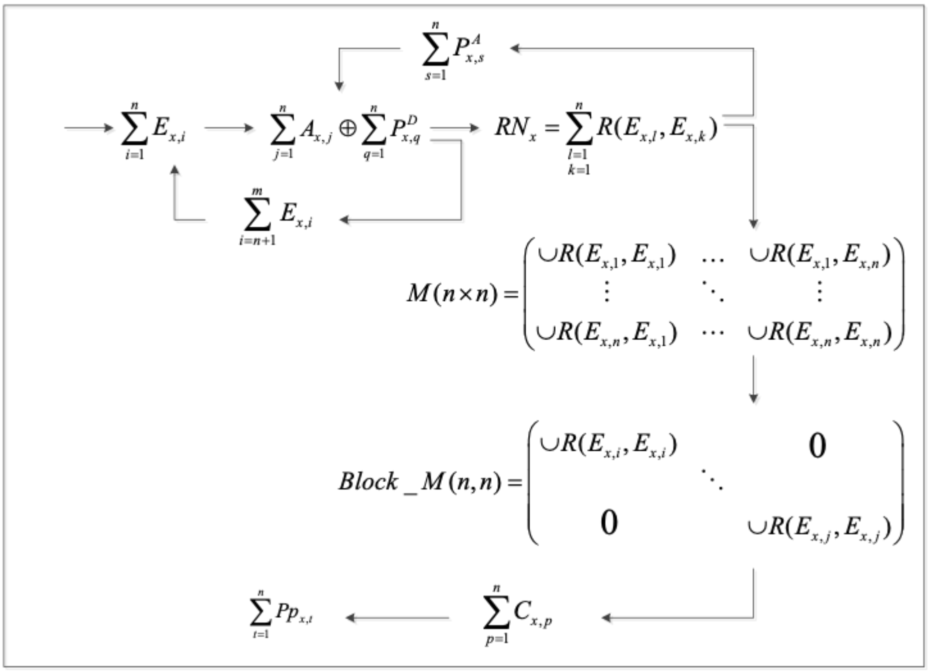


Figure 3.11. Steps of getting from entities to propositions

building a new theory. Irrelevant propositions (vague or out of context) were directly deleted and were not considered in further steps of the methodology. Relevant and partially relevant (those that showed a certain level of implication but were not concrete enough) propositions were kept so we could study their further implications in the theory fusion step. Up to this point in the methodology, all theories had been studied individually.

3.3.5.7. Fusion of component theories

Theory fusion was the most challenging step of the process, since we were no longer studying component theories separately. This process started from the combined set of entities and relationships from the *N* original theories. The objective was to merge them semantically into one robust theory that went beyond what component theories were able to cover individually. The enormous amount of information (theories, entities, and relationships) present at the beginning of the theory fusion made the process time-consuming and complex. Fusion was a delicate process because it was not only syntactic but also semantic in nature. This process featured seven steps:

- **Step 1 – Entities combination:** The objective of this step was to group lists of entities extracted from individual theories into one final list. The redundancies (repeated entities and synonymous entities) were removed and only one version was kept. These so-called common entities were represented as $E_{c,i}$ (where i is the running index of the common entity). In the end, a final unified list of entities was compiled. At this point we observed links created by the common entities between theories.
- **Step 2 – Axiomatization:** The objective of this step was to integrate all axioms and postulates of component theories into one list. In the case of redundant or synonymous axioms and postulates, only one version was kept in the final list.
- **Step 3 – Relationships networks combination:** This step systematically combined previously constructed relationships networks of the component theories into a compound network using the final list of entities and relationships (axioms and postulates). As can be seen in Figure 3.12, theories were linked to each

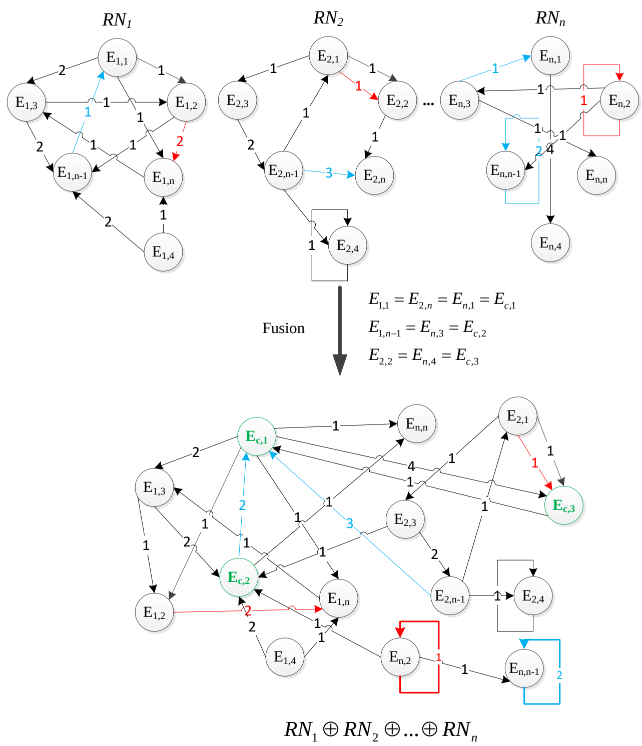


Figure 3.12. Relationships network established over fused theories

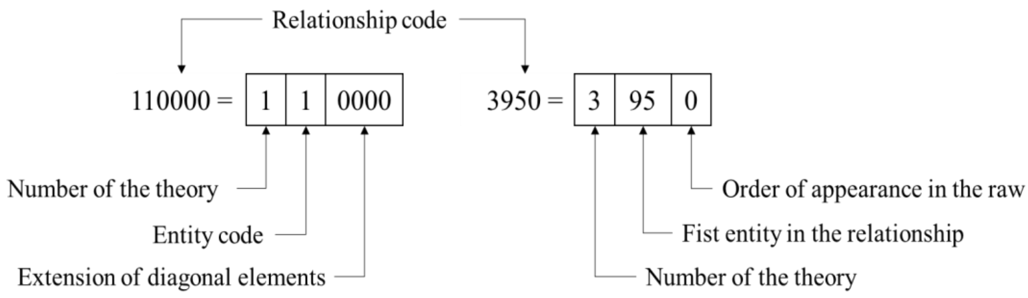


Figure 3.13. Example of coding used for fused theories

other via common entities. This form of bridging between theories was augmented by creating extra relationships when possible between not-linked or distant entities from the same theory or from different theories.

- **Step 4 – Matrix decomposition:** This step was similar to the previous step but used the final list of entities and relationships. Here we first observed that the order of entities had shifted due to the common entities, emphasizing the already seen bridging between theories.
- **Step 5 – Matrix rearrangement:** In this step, we started by coding the new groups of relationships between entities represented in the decomposed matrix. The method of coding was the same as the one presented in Section 3.3.5.5 (Figure 3.13 illustrates how this method was applied in the current step). The common entities followed the same procedure. This means that the "theory code" was chosen based on the first appearance of the theory. The generated codes replaced the relationships in the matrix. The matrix was then processed in Matlab to generate the block-matrix, which is used in the next step, and, once again, to replace the numerical codes with the textual ones.
- **Step 6 – Extraction of propositions:** In this step, we analyzed clusters reflected by the block-matrix. This procedure followed the same logic as the one presented previously. Only relevant and partially relevant clusters were kept, based on their implications and their possible contextualization. This step required particular attention, since combining diverse theories semantically into a specific context was a challenging task. After we had derived propositions from the selected clusters, we revisited them to check whether all aspects of the domain of interest were present and whether further theories needed to be investigated.
- **Step7 – Textual formulation:** In this step, we transcribed the final list of propositions derived for the target theory into a set of rationally implied requirements. These requirements captured the implications of the basis of the new theory. We filtered and ordered the final set of requirements based on what was needed for the new theory. Those requirements that did not add novelty or implications were removed.

3.3.5.8. *Validation of the new theory*

The goal of the validation study was to validate the proposed ATF as a new methodology for building theories. To evaluate and check the consistency of the obtained theory, we tested ATF in a demonstrative application case in design, where combining theories remains a challenge. In doing so, we considered several criteria. The first involved judging whether the combined theories explained, communicated, and demonstrated more than what had been reflected by each component theory individually. The second criterion consisted of specifying whether the outcomes of the merging provided a sufficient knowledge platform. Finally, the last criterion consisted of checking whether the target theory was operational and the set of requirements could be used in a real-life application. To validate the use of the ATF methodology, we considered three aspects: (i) the usability of the ATF methodology in a specific context, (ii) the overall effectiveness of the ATF methodology, and (iii) the success of the ATF methodology at building a robust knowledge platform.

3.3.6. Relevance criteria and conceived limitations

To determine whether the theory fusion approach could be used in different application contexts, several relevance criteria were assessed:

- **The relevance of the set of axioms derived:**
  The objective was to determine if the extracted axioms from the component theories were coherent and sufficient to deliver knowledge in the application context.
- **The relevance of the set of postulates determined:**
  The objective was to analyze whether the derived postulates, but more importantly the auxiliary postulates, formed the first step towards contextualization through the added relationships that filled the gaps between disconnected entities of the theories. We also checked whether postulates were coherent and sufficient and whether auxiliary postulates connected the theory to the concrete application case.
- **The specification of the context:**
  The objective was to check to what extent the original context of the component theories and the new context of the theories (the target application cases) resembled each other, overlapped, or were disconnected.
- **The relevance of the propositions:**
  The objective was to judge whether the knowledge communicated by the fused theories served its purpose, was more useful than the knowledge conveyed by the individual theories, and filled a gap in the literature concerning theories.
- **The choice of the component theories:**
  The objective was to determine whether selected theories were sufficient, covered all research aspects, and provided a robust knowledge platform for theory building.

During the application process, some weaknesses of the ATF methodology were observed, which were related to its practical application rather than to its theory fusing capabilities. The observed weak points were (i) time consumption, (ii) error detection,

(iii) automation, and (iv) application. Actually, these are procedural limitations, which are further explained below:

- The methodology requires a large number of manual procedures, which are difficult to manage when there are a large number of theories and/or theories with highly rich content, as these make the procedure more time-consuming and increase the chances of human error.
- The application of the methodology involves many decision points. Incorrect observations and interpretations may have an unfavorable impact on the outcomes, and unrecognized errors may lead to inappropriate results. The recognition of this can also cause a potentially substantial delay concerning the point in time when the mistake was made.
- Although beneficial, the proposed approach is still in its infancy with regard to its level of computational support (and automation). This limited support contributes to the two issues and limitations mentioned above. On the other hand, proper computer support can solve many technical issues and can aid even semantic reasoning.

The methodology was tested in one demonstrative application, which involved the merging of five theories. It was used to explore means about the ATF methodology. Its goal was to test the correctness of ATF and use it to derive knowledge in the particular case of DATs development for white goods enhancement by product designers. This application proved the feasibility and usability of the procedure and helped generate information about possible enhancements.

3.4. Deductive study: Application of the axiomatic theory fusion methodology in theory synthesis

3.4.1. Exposition of the problem and research objective

As do other professionals, developers of white goods need data analytics tools that are tailored to their problems, needs, knowledge, and expertise. Theories that support the development of traditional user software and data analytics tools have proven to be insufficient in this context. Consequently, there was a need to provide proper theories that describe the new tools and explain what functions and computation are necessary. Our preliminary research determined that of theories of this kind are scarce. One novel approach to deriving comprehensive supporting theories is semantic fusion of relevant component theories. The principles of this approach, the ATF methodology, were reported in Section 3.3. In this section, ATF is applied to derive a theory for data analytics toolbox development.

This demonstrative application involved theory development in a specific design context. The overall objective was to develop a comprehensive theory supporting the conceptualization and implementation of an SDATB for functional and embodiment design of white goods based on aggregation and exploration of MoLD. The desired theory should provide an ontological description of what exists or what should exist in the development of an SDATB. The theory should specify crucial aspects of SDATB manifestation, which can include behavioral and functional expectations and opportunities of the SDATB.

3.4.2. Theoretical considerations and choice of relevant component theories

Our reasoning for this deductive study is represented in Figure 3.14. It illustrates an overall process flow of enhancement of a particular family of products (white goods) by product developers (product designers) who are not data analysts. According to our conceptualization, these product developers used various data analytics tools to generate data about the MoL of products and converted these data into knowledge that served as the basis of idea generation for product enhancement. The most favorable enhancement options should be chosen by decision-making. Several important methodological connections among the elements of the overall process flow could be identified.

Assuming that a family of products was given, the first methodological connection (T_1) concerned the generation of data, information, and knowledge about the MoL of products by product developers not specialized in data analytics, using advanced data analytics tools, technologies, and assets. The second methodological connection (T_5) related to the conversion of product related data, information, and knowledge into generic product enhancement options and concrete concepts. The third methodological connection (T_4) concerned the methods designers could use to extract patterns and handle data while considering their plans and the strategy of the company. The fourth methodological connection concerned the operationalization of ideas, concepts, and analogies in the framework of a strategy plan, vision, and/or concrete enhancement and new design options and how the system could be architected (T_3). Finally, the essence of the fourth methodological connection was decision-making about changes to be introduced to optimally enhance the elements or the whole of the given product family (T_2).

In the process of enhancing the target family of products (white goods), several epistemological connections were identified. In fact, these dependencies could also be interpreted from a methodological point of view. The most influential ones were as follows:

- The generation of data, information, and knowledge about the MoL of products by product developers, not specialized in data analytics, using advanced data analytics tools, technologies, and assets.

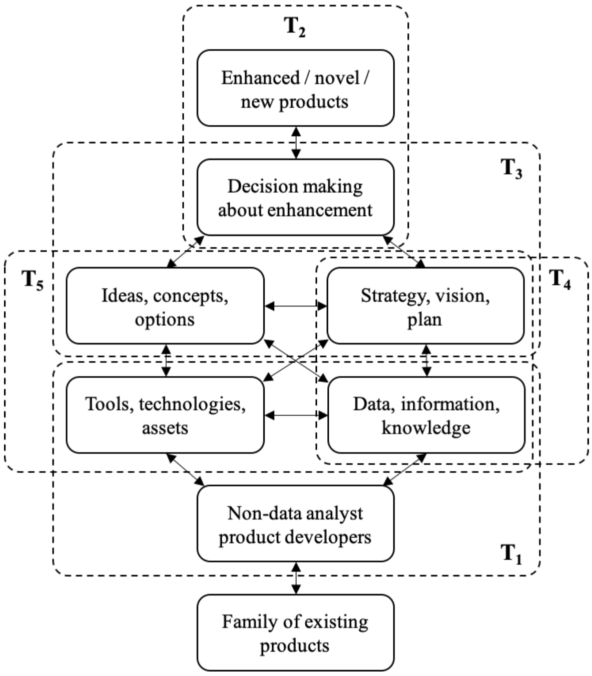


Figure 3.14. Reasoning model of the deductive study

- The conversion of product-related data, information, and knowledge into generic product enhancement options and concrete concepts.
- The operationalization of ideas, concepts, and analogies in the framework of a strategy, plan, or vision and/or concrete enhancement options and new design options.
- Decision-making about changes to be introduced to optimally enhance the elements or the whole of the given product family.

These forms of methodological connection were nontrivial and entailed the need for theories that explain the relationships from both epistemological and methodological points of view.

Based on previous knowledge, the theories to be fused were chosen based on the following assumption: "*a robust and comprehensive conceptual basis for a knowledge platform for a next generation data analytics toolbox for white goods designers needs a combination of many composite theories about (i) explicit needs of designers, (ii) issues of interoperability, (iii) principles of decision-making, (iv) evolution of data analytics, and (v) enabling technologies*." After investigating existing theories, we selected the following theories:

- **T_1: Theory concerning the needs of designers**
  This theory captured the needs of white goods designers related to data analytics tools through a set of inductively generated requirements that they should fulfill when used in this context. This theory described designers needs based on their relationships with data analytics tools. The theory was derived from a study that investigated the current situation of design improvement using data analytics tools and packages, based on practical information from designers [86].
- **T_2: Theory describing advanced technological enablers**
  The enabling technologies theory provided knowledge about how software and cyber-physical-system tools could be exploited as enablers [87].
- **T_3: Theory explaining the evolution of data analytics**
  The evolution of data analytics theory provided knowledge on the methods to be used by designers and the means for patterns extraction and for handling MoLD [88].
- **T_4: Theory of combined creative problem-solving and decision-making**
  The decision-making theory provided knowledge on principles and methods of optimal decision-making. Two relevant theories were chosen that described the methodological and epistemological relationships between preliminary knowledge generation, courses of action, effective solution development, and context-driven robust decision-making. One theory is the theory of creative problem solving, which relies on the concept of proactive decision support [89], and the other is the theory of creative decision-making [90].

- **T_5: Theory of functional and structural interoperability**
  Interoperability theory helped determine the architecture of the system as well as the functional design and the structural arrangement of components [91].

3.4.3. Axiomatic discretization of component theories

As proposed by the ATF methodology, we filtered the five previously selected theories and kept only statements with implications to the application context. Those statements were sentences composed of entities and the relationships between them. The total number of entities extracted from T_1 was 129. From T_2, 218 entities were extracted. From T_3, 174 entities were extracted. From T_4, 78 entities were extracted. Finally, from T_5, 79 entities were extracted. Table 3.8 shows a sample of entities derived from the five component theories. (Appendix 2 presents the detailed decomposition of T_1).

After determining entities and statements extracted from the textual formulations of theories, we used the method of axiomatization to determine axioms and postulates. Applying this method for each theory, we generated the following:

- T_1: 78 axioms, 10 derived postulates, and 89 auxiliary postulates
- T_2: 153 axioms, 7 derived postulates, and 134 auxiliary postulates
- T_3: 141 axioms, 3 derived postulates, and 113 auxiliary postulates
- T_4: 67 axioms, 17 derived postulates, and 36 auxiliary postulates
- T_5: 57 axioms, 7 derived postulates, and 29 auxiliary postulates

3.4.4. Sematic and visual capturing of relationships

The sets of entities, axioms, and postulates were used to visualize relationships within each theory. Figure 3.6 is a part of the relationships network of theory 1 (for clear visibility of relationships), showing the captured relationships between entities. It can be seen that auxiliary postulates (blue arrows) connect numerous disconnected and distant entities.

Table 3.8. Examples of entities from the five component theories

Theory	Entity code	Denomination	Entity code	Denomination	Entity code	Denomination
T_1	$E_{1,1}$	Data analytics tool	$E_{1,2}$	Knowledge	$E_{1,3}$	Product
T_2	$E_{2,1}$	Big data	$E_{2,2}$	Volume	$E_{2,3}$	Huge data amount
T_3	$E_{3,1}$	Data analytics	$E_{3,2}$	Data collection	$E_{3,3}$	Data organization
T_4	$E_{4,1}$	Design problem	$E_{c,3}$	Product	$E_{4,3}$	Functional requirement
T_5	$E_{5,1}$	Human–system interaction	$E_{5,2}$	Level of interaction	$E_{5,3}$	Domain of interaction

Table 3.9. Examples of coding of the five component theories

Theory	Code	Relationships	Code	Relationships	Code	Relationships	Code	Relationships
T_1	243	$A_{1,16}$; $A_{1,17}$; $A_{1,18}$	244	$A_{1,19}$; $A_{1,20}$; $A_{1,21}$	250	$P^A_{1,18}$	260	$P^A_{1,19}$
T_2	361	$P^D_{2,6}$	362	$A_{2,34}$	363	$P^A_{2,17}$	370	$P^A_{2,20}$
T_3	-911	$P^A_{3,96}$	-912	$A_{3,124}$	920	$A_{3,130}$	921	$A_{3,131}$
T_4	46000	$A_{4,104}$; $A_{4,105}$; $A_{4,108}$	47000	$A_{4,89}$	49000	$A_{4,99}$	51000	$P^D_{4,10}$
T_5	10	$A_{5,1}$	11	$A_{5,2}$	12	$A_{5,3}$	13	$A_{5,4}$

For another dimension of visualization, we carried out the matrix decomposition of the five theories to facilitate movement toward a semantic capturing of relationships. In this representation, entities were rows and columns of the matrix (symmetric matrix), and relationships connected the entities. The matrixes generated for all theories were large, which made it difficult to capture and include an illustration of this step (reduced representation is included in the next section).

To perform the matrix rearrangement, we had to first use the coding method to code the relationships included in the matrix before putting it into Matlab for processing and rearrangement. The coding method has no semantic association with the components of the purpose (Matlab usage). Table 3.9 lists examples of the codes used for the five theories. The coding followed the principles reported in the illustrated examples in Figure 3.8. After replacing relationships in the matrixes with numerical codes, we entered the obtained numerical matrixes into Matlab for rearrangement.

The result was a block-matrix for each theory, composed of several clusters. These matrixes were again put into Matlab to replace the codes with the original symbols to visualize all relationships and facilitate the analyses. In the results of this procedure, 30 clusters were distinguished for T_1, 54 clusters for T_2, 40 for T_3, two large clusters for T_4,

Table 3.10. Examples of propositions from the five component theories

Theory	Cluster number	Proposition
T_1	26	Designers need procedural reasoning and case-based reasoning
T_2	38	Traditional and mathematical modeling do not solve complex real-world data driven problems
T_3	40	Data must be formatted to be suitable for data mining and subsequent analysis
T_4	2	Context makes knowledge-based systems reliable
T_5	5	Intelligent-based System-human interaction, proactivity and awareness are to be considered in the case of intelligent systems

and six clusters for T_5. Note that the matrixes obtained in the matrix decomposition process were huge and therefore could not be inserted in this dissertation (i.e. the representation used in the next section has been reduced).

The following step was the extraction of meaningful propositions out of the identified blocks. By analyzing the clusters of each theory separately, we identified 15 relevant, 1 partially relevant, and 14 irrelevant clusters in T_1; 41 relevant, 4 partially relevant, and 9 irrelevant clusters in T_2; and 26 relevant, 7 partially relevant, and 7 irrelevant clusters in T_3. The two clusters of T_4 were both relevant, and T_5 contained 5 relevant clusters and 1 irrelevant cluster. By analyzing relevant and partially relevant clusters of each theory, we derived 34 propositions from T_1, 76 propositions from T_2, 45 propositions from T_3, 27 propositions from T_4, and 12 propositions from T_5. Examples of these propositions are given in Table 3.10. Analyzing the implications of the propositions formulated in the application context revealed that they were all directly or indirectly linked to the research phenomenon. For this reason, no filtering of the propositions was done in this step.

3.4.5. Syntactic processing and merging of component theories

Based on the ATF methodology, our first step in merging theories was to combine the lists of entities of component theories and then merge them into one list with no redundancies. Our final list contained 574 entities, including 81 common entities, with 4 of them common to four theories, 12 common to three theories, and 65 common to two theories. Examples of common entities can be seen in Table 3.11. The second step was the axiomatization of the combined theories, taking into consideration the final list of entities. This step required less effort; since axioms and postulates were already established and written in their final format, only duplications needed to be removed. From the final list of relationships, four repeated axioms and 13 repeated auxiliary postulates were deleted, resulting in a list of 924 relationships, including 492 axioms, 44 derived postulates, and 388 auxiliary postulates.

Our third step was to establish a relationships network representing all theories that considered the final list of entities and relationships. Figure 3.15 shows a representative (but simplified) part of the relationships network of the combined theories. The common entities (shown in green) and the auxiliary postulates (shown in blue) connect the component theories to each other. To facilitate capturing the semantics of the relationships, this visual representation of the relationships network was converted into a matrix representation. From this representation, the statements concerning the

Table 3.11. Examples of common entities among the five component theories

Entity code	Common entities	Denomination	Entity code	Common entities	Denomination
$E_{c,4}$	$E_{1,9} = E_{2,34} = E_{3,9} = E_{4,18}$	Data	$E_{c,55}$	$E_{2,190} = E_{3,155} = E_{4,16} = E_{5,11}$	System
$E_{c,1}$	$E_{1,1} = E_{2,208} = E_{3,16}$	Data analytics tools	$E_{c,74}$	$E_{3,174} = E_{5,36}$	Technology

Table 3.12. Examples of coding from the fused theories

Code	Relationships	Code	Relationships	Code	Relationships	Code	Relationships
110000	$A_{1,36}$; $P^{D}_{1,5}$; $P^{D}_{1,6}$	-2342	$P^{A}_{3,109}$	3293	$A_{5,22}$; $A_{5,23}$	5780	$P^{D}_{5,3}$

particular entities could be grouped and further analyzed.

The size of the connectivity matrix was 574 × 574 cells. It contained 878 defined relationships (including 11 elements of the main diagonal, which expressed self-reflecting relationships). As an illustration, a simplified version of the matrix is shown in Figure 3.16, where the particular entities (shown in black and in bold fonts), the common entities (in green), the axioms (in light black), the derived postulates (in red), and the auxiliary postulates (in light blue) are all included. The 0 values that can be seen indicate the non-existence of relationships between the entities. Our fifth step in the theory fusion started with coding the relationships within the matrix using numerical values. The same method of coding presented in Section 3.3.5.5 was applied and produced several differences in the obtained codes, since the order of the appearance of the entities had changed because of the removal of redundancies and the order of appearance of common entities. Table 3.12 shows examples of the established codes.

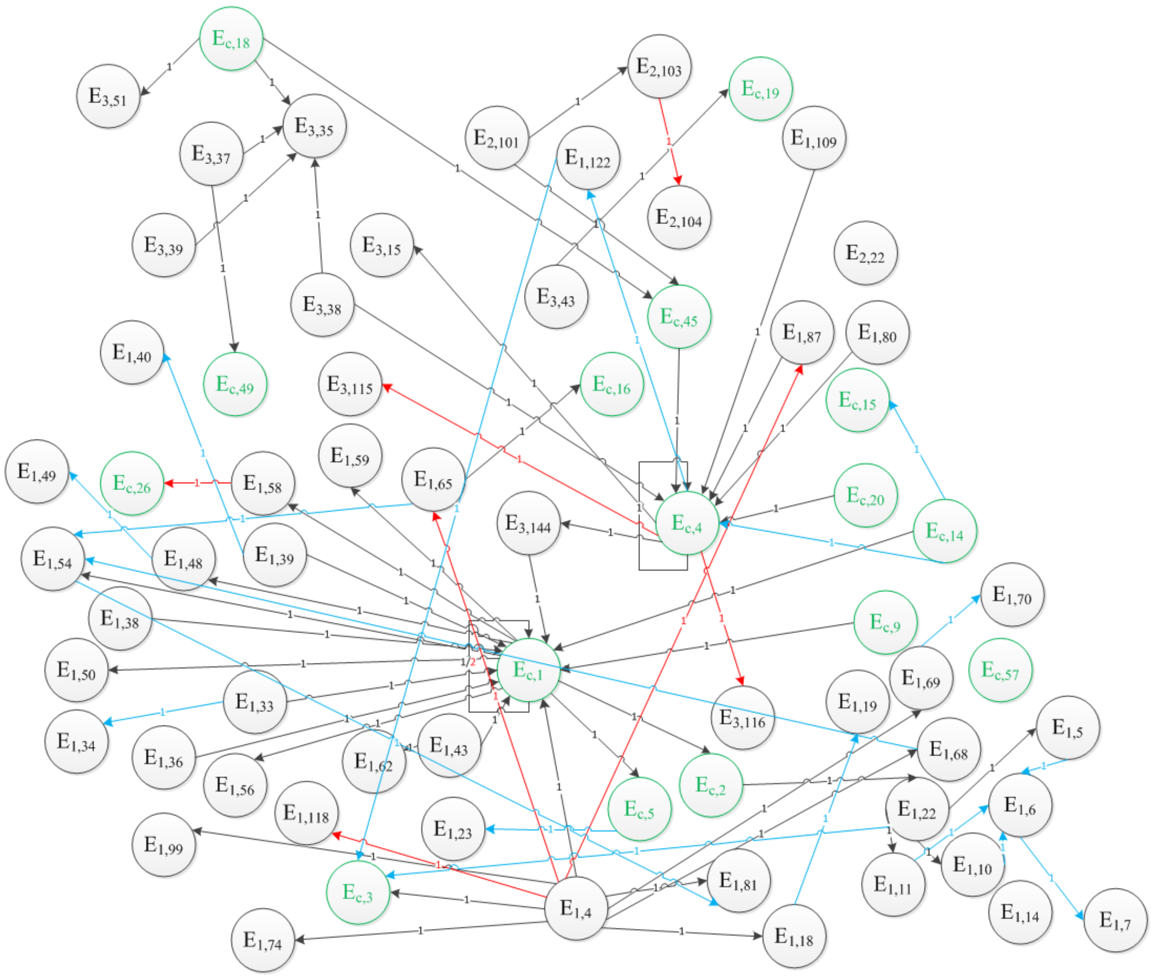


Figure 3.15. Part of the relationships network between the five combined theories

	Ec,1	Ec,2	Ec,3	…	E1,24	E1,25	…	Ec,39	E2,20	…	E3,6	E3,7	E3,8	…	E4,1	E4,3	E4,4	…	E5,77	E5,78	E5,79
Ec,1	$A_{1,36}$; $P^{D}_{1,5}$; $P^{D}_{1,6}$	$A_{1,1}$	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
Ec,2	$A_{1,1}$	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
Ec,3	0	0	0	…	$P^{A}_{1,17}$	0	…	0	0	…	0	0	0	…	$A_{4,1}$	$A_{4,2}$	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E1,24	0	0	$P^{A}_{1,17}$	…	0	$A_{1,13}$	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
E1,25	0	0	0	…	$A_{1,13}$	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Ec,39	0	0	0	…	0	0	…	0	$A_{2,18}$	…	0	0	0	…	0	0	0	…	0	0	0
E2,20	0	0	0	…	0	0	…	$A_{2,18}$	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E3,6	0	0	0	…	0	0	…	0	0	…	0	$A_{3,7}$	$A_{3,8}$	…	0	0	0	…	0	0	0
E3,7	0	0	0	…	0	0	…	0	0	…	$A_{3,7}$	0	0	…	0	0	0	…	0	0	0
E3,8	0	0	0	…	0	0	…	0	0	…	$A_{3,8}$	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E4,1	0	0	$A_{4,1}$	…	0	0	…	0	0	…	0	0	0	…	0	$P^{D}_{4,8}$	0	…	0	0	0
E4,3	0	0	$A_{4,2}$	…	0	0	…	0	0	…	0	0	0	…	$P^{D}_{4,8}$	0	0	…	0	0	0
E4,4	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E5,77	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
E5,78	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	$P^{D}_{5,3}$
E5,79	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	$P^{D}_{5,3}$	0

Figure 3.16. Simplified representation of the matrix regrouping the five theories

As shown in Figure 3.17, these codes were manually inserted in the matrix, which was entered into Matlab and transformed to a block-matrix containing 95 clusters. As a simplified visualization, Figure 3.18 shows one of the obtained clusters (C_{22}, where 22 is the order of appearance of the cluster in the matrix from right to left). Figure 3.19 represents the same cluster after Matlab has replaced the codes with the original symbolic representations (to provide a clear visualization of the relationships). The remaining two steps of the theory fusion are detailed in Sections 3.4.6 and 3.4.7, because they needed particular attention.

3.4.6. Deriving propositions based on units of resultant theory

The total set of clusters contained 40 relevant, 13 partially relevant, and 42 irrelevant clusters. For irrelevant clusters (vague information or no implication for the application context), a large distance between relationships was identified. This indicates that within these clusters relationships were separated by many 0s, which meant they were not

	Ec,1	Ec,2	Ec,3	…	E1,24	E1,25	…	Ec,39	E2,20	…	E3,6	E3,7	E3,8	…	E4,1	E4,3	E4,4	…	E5,77	E5,78	E5,79
Ec,1	110000	110	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
Ec,2	110	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
Ec,3	0	0	0	…	132	0	…	0	0	…	0	0	0	…	135	136	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E1,24	0	0	132	…	0	1240	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
E1,25	0	0	0	…	1240	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Ec,39	0	0	0	…	0	0	…	0	2180	…	0	0	0	…	0	0	0	…	0	0	0
E2,20	0	0	0	…	0	0	…	2180	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E3,6	0	0	0	…	0	0	…	0	0	…	0	360	361	…	0	0	0	…	0	0	0
E3,7	0	0	0	…	0	0	…	0	0	…	360	0	0	…	0	0	0	…	0	0	0
E3,8	0	0	0	…	0	0	…	0	0	…	361	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E4,1	0	0	135	…	0	0	…	0	0	…	0	0	0	…	0	410	0	…	0	0	0
E4,3	0	0	136	…	0	0	…	0	0	…	0	0	0	…	410	0	0	…	0	0	0
E4,4	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
E5,77	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	0
E5,78	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	0	5780
E5,79	0	0	0	…	0	0	…	0	0	…	0	0	0	…	0	0	0	…	0	5780	0

Figure 3.17. Matrix coded to be used for the rearrangement

communicating valuable information. These irrelevant clusters were not used to derive propositions. This did not affect the relevance of the methodology, since clusters were not identical in terms of the numbers of included relationships. For example, one relevant cluster can contain more relationships than a group of 10 irrelevant clusters. From the relevant and partially relevant clusters, 82 propositions were constructed, of which 77 were relevant and 5 only partially relevant.

	…	E1,24	E2,25	E4,3	E1,103	…
…	…	…	C22	…	…	…
E1,31	…	1244	0	0	0	…
E1,29	…	1243	0	0	0	…
E1,27	…	1242	0	0	0	…
E1,26	…	1241	0	0	0	…
E1,25	…	1240	0	0	0	…
E2,28	…	0	2251	0	0	…
E1,109	…	0	0	0	11035	…
E1,108	…	0	0	0	11034	…
E1,107	…	0	0	0	11033	…
E1,105	…	0	0	0	11031	…
…	…	…	…	…	…	…

Figure 3.18. Matrix rearrangement using Matlab

Below, the already presented C_{22} cluster is analyzed to illustrate the process of deriving propositions into the context of data analytics tools for enhancement of white goods by designers:

- The relationships contained in C_{22} were as follows:

$A_{1,19}$: (Smart products)$_{1,24}$ [collect their] (use circumstances)$_{1,31}$

$A_{1,20}$: (Smart products)$_{1,24}$ [communicate their] (use circumstances)$_{1,31}$

$A_{1,21}$: (Smart products)$_{1,24}$ [reason with their] (use circumstances)$_{1,31}$

$A_{1,16}$: (Smart products)$_{1,24}$ [collect their] (operational states)$_{1,29}$

$A_{1,17}$: (Smart products)$_{1,24}$ [communicate their] (operational states)$_{1,29}$

$A_{1,18}$: (Smart products)$_{1,24}$ [reason with their] (operational state)$_{1,29}$

$A_{1,15}$: (Smart products)$_{1,24}$ [incorporate] (self-management capabilities)$_{1,27}$

	…	E1,24	E2,25	E4,3	E1,103	…
…	…	…	C22	…	…	…
E1,31	…	$A_{1,19}$; $A_{1,20}$; $A_{1,21}$	0	0	0	…
E1,29	…	$A_{1,16}$; $A_{1,17}$; $A_{1,18}$	0	0	0	…
E1,27	…	$A_{1,15}$	0	0	0	…
E1,26	…	$A_{1,14}$	0	0	0	…
E1,25	…	$A_{1,13}$	0	0	0	…
E2,28	…	0	$A_{2,25}$	0	0	…
E1,109	…	0	0	0	$A_{1,75}$	…
E1,108	…	0	0	0	$A_{1,74}$	…
E1,107	…	0	0	0	$A_{1,73}$	…
E1,105	…	0	0	0	$A_{1,71}$	…
…	…	…	…	…	…	…

Figure 3.19. Matrix coded using Matlab

$A_{1,14}$: (Smart products)$_{1,24}$ [incorporate] (self-adaptation capabilities)$_{1,26}$

$A_{1,13}$: (Smart products)$_{1,24}$ [incorporate] (self-learning capabilities)$_{1,25}$

$A_{2,25}$: (Sentiment analysis) $_{2,25}$ [identifies] (user's opinion)$_{2,28}$

$A_{1,75}$: (System intellect)$_{1,103}$ [is provided by] (system adaptation capabilities)$_{1,109}$

$A_{1,74}$: (System intellect)$_{1,103}$ [is provided by] (strategy development)$_{1,108}$

$A_{1,73}$: (System intellect)$_{1,103}$ [is provided by] (situation awareness)$_{1,107}$

$A_{1,71}$: (System intellect)$_{1,103}$ [is provided by] (system learning mechanisms)$_{1,105}$

Table 3.13. Examples of propositions of different levels

Propositions		
Main functions (1st level)	Sub-functions (2nd level)	Sub-sub-functions (3rd level)
Learning	(Like smart products) The SDATB needs to incorporate self-learning capabilities	The SDATB needs to incorporate reinforcement learning, density estimation, and dimensionality reduction
	(Like smart products) the SDATB uses system intellect	The SDATB system intellect is provided by learning mechanisms
Procedural reasoning	(Like smart products) the SDATB needs to collect, communicate, and reason with its use circumstances and operational state	An SDATB needs to include smart semantics to extract meaning from collected data
		An SDATB needs to include ontologies and reasoning engines for semantic interpretations
Data types and characteristics	The SDATB identifies the user's opinion	The SDATB allows sentiment analysis in order to identify user's opinion

where $(Text)_{x,i}$ is the textual formulation of the entity, x is the order of the theory, and i is the order of appearance of the entity in the theory, and [*Text*] is the type of relationship between two entities.

By combining the axioms semantically and putting them into the context of the study, we derived a set of propositions. The natures of these propositions fell into four categories: (i) requirement, (ii) descriptive, (iii) explanatory, and (iv) control content. Below is a sample of obtained propositions of different natures (the number represents the order in which the propositions were formed):

Proposition$_{11}$: "(Like smart products) the SDATB incorporates self-learning, self-management, and self-adaptation capabilities." This proposition is of a descriptive nature.

Proposition$_{12}$: "(Like smart products) the SDATB collects, communicates, and reasons with its use circumstances and operational state." This proposition is of a descriptive nature.

Proposition$_{13}$: "(Like smart products) the intellectualization of the SDATB is provided by system learning mechanisms, situation awareness, strategy development, and system adaptation capabilities." This proposition is of a control content nature.

Proposition$_{14}$: "The SDATB allows sentiment analysis in order to identify user's opinion." This proposition is of an explanatory nature.

- Since this cluster did not contain partially relevant propositions, here are some examples from other clusters:

  "The SDATB should propose creative solutions to solve difficult design problems." This proposition is of a requirement nature.

  "The SDATB is to be affordable." This proposition is of a requirement nature.

  "The SDATB is user-friendly." This proposition is of a descriptive nature.

  "The SDATB improves usability." This proposition is of a descriptive nature.

The general set of propositions was filtered, and only the relevant propositions were kept. The second step was to check if all propositions belonged at the same level. This checking distinguished two levels of propositions from the general to the particular that could be put under a main general category (example: decision-making, learning, interfacing, etc.). This result was unexpected. This categorization is clarified in Table 3.13. The next step was to convert the functionalities (propositions) into requirements.

3.4.7. Transfer of the propositions into a narrative description

The final set of propositions was dedicated to the context of developing a new-generation data analytics toolbox that goes beyond individual tools and covers white goods designer's expectations. It described the expectations and the basis for the toolbox implementation requirements. Some of the requirements (algorithms and methods) were clearly formulated in the clusters and were evident without further investigation. We had to further investigate the remaining requirements to determine their implications for the construction of the toolbox. Examples of the obtained requirements are presented below:

- Reinforcement learning, density estimation, and dimensionality reduction could be included in the toolbox.
- System learning mechanisms, situation awareness, strategy development, and system adaptation capabilities could be part of the toolbox.
- Spatial and temporal correlation within the SDATB could be modeled using a predictive deep convolutional neural network.
- Large-scale data within the SDATB could be modeled using support vector machines, naive Bayes, or logistic regression.
- The SDATB could learn from experiential data using a back propagation algorithm.

The outcomes of the ATF application were represented moving from the general category of functionalities (general level of propositions) to the final list of requirements. To easily visualize, organize, and analyze the outcomes, we systematically arranged them as represented in Table 3.14.

3.4.8. Applicability validation of the axiomatic theory fusion methodology

To validate the ATF methodology, we decided to use the principles of applicability validation. ATF was used in the particular case of developing a theory to build a next generation data analytics toolbox. In a perfect situation, other validation methods could be realized – for example, comparing ATF to similar methodologies as another means for validating its performance. The novel and unique aspect of our methodology made the all-embracing methodological validation impossible. The current validation method is used to evaluate the criteria cited in Section 3.3.6. To this end, we posed the following questions:

Was the set of derived axioms and postulates relevant?

Were the propositions contextualized, and did they serve their purpose?

Did the chosen theories provide the needed knowledge platform?

Deriving axioms and postulates from the five component theories was a systematic syntactic procedure. This made their aggregated set logical, presenting one-to-one relationships and containing no bias or interpretation. This step was easily feasible, with no contradictions or ambiguities. As used in theory fusion, the common entities facilitated the contextualization of axioms and postulates by linking them via more than one common entity. This was completed by creating auxiliary postulates that developed more links between theories and connected disconnected entities. The graphical representation showed bridging between theories, resulting in a densely connected web within the research context.

By evaluating the propositions derived from the set of relationships supported by axioms and postulates, the result was satisfying, since it did not present contradictions and formed a knowledge platform about the ideation of a new-generation data analytics toolbox. This meant also that the set of axioms and postulates was relevant. The obtained requirements connected the specification of the new-generation toolbox and its ideation and conceptualization. Considering all findings, it is possible to conclude that chosen theories were relevant and allowed the establishment of the needed knowledge platform for the application case.

Table 3.14. Representation of the outcomes of the axiomatic theory fusion

<table>
<tr><td>L_1F_X</td><td>$L_2F_{x,y}$</td><td>$L_3F_{x,y,z}$</td><td>R_i</td></tr>
<tr><td rowspan="5">L_1F_1</td><td rowspan="3">$L_2F_{1,1}$</td><td>$L_3F_{1,1,1}$</td><td rowspan="4">R_1</td></tr>
<tr><td>$L_3F_{1,1,2}$</td></tr>
<tr><td>$L_3F_{1,1,3}$</td></tr>
<tr><td>$L_2F_{1,2}$</td><td>$L_3F_{1,2,1}$</td></tr>
<tr><td>$L_2F_{1,3}$</td><td>$L_3F_{1,3,1}$</td><td>R_2</td></tr>
</table>

L_1F_X is a main function x; $L_2F_{x,y}$ is a first-level sub-function y derived based on the main function x; $L_3F_{x,y,z}$ is a second-level sub-sub-function z derived from y; and R_i is a requirement of running index i.

The use of the ATF methodology in this application demonstration case proved that the new methodology for axiomatic fusion of qualitative engineering theories is valid in real-life applications. Using such a methodology can provide surprising and unexpected results. In the current case, these results include those regarding (i) the grouping of propositions, (ii) providing practical requirements, and (iii) the rule that "*the more relationships were closer to each other the more they communicated and the more they contributed to forming a knowledge platform (and vice versa)*."

The objective of applying ATF in a particular data analytics design context was to develop a knowledge platform for a next generation SDATB for white goods designers. The outcomes exceeded the objective by providing a multilevel set of functionalities and requirements to be implemented for a next generation SDATB. In this particular example, the ATF methodology was relevant.

3.4.9. Reflections on the axiomatic theory fusion and its application

The motivation for this research was to overcome the lack encountered in developing new multidisciplinary theories. The methodologies for theorizing proposed in the literature cannot be used directly in some research areas in which more than one aspect needs to be considered. The complications of using existing approaches became more persistent in the presence of semantically distant aspects. The methodology of ATF was able to fill in some of these gaps. ATF made theories that were insufficient individually (in some contexts or when used alone) more valuable and insightful when they were combined. It was challenging to develop a methodology using the bases of a mathematical methodology (axiomatization principles) and converting them to be used in a design context. The challenge was not only in the mathematical nature of the components (axioms and postulates) but also in the manual work done in major parts of the approach. This was a beneficial but time-consuming procedure because of the high level of precision and focus required in all steps to avoid mistakes that risked becoming apparent in later stages. Representation, such as developing the relationships network, was one of the longest steps due to the large number of entities included in the studied application case.

The methods identified and operationalized in the ATF methodology included both manual and computer-aided methods and techniques. The manipulations can be challenging, difficult, and time-consuming in certain complex application cases that involve fusing a large number of theories. Nonetheless, these manual manipulations could not be fully automated given the need for human comprehension, semantic interpretation, logical reasoning, reductionist decomposition, consistency checking, and compliance testing. Other limitations of ATF were present in the axiomatization of certain theories that could be performed in a straightforward manner but only after structural and representational transformations, which can lead to the issue of theory congruence. In ATF, human interpretation and intuition were indispensable concerning the proposed procedures, methods, and instruments. Their experience level would significantly influence the efficiency and correctness of knowledge processing (including decisions about the axiomatic primitives, semantic relations, assignment of clustering codes, manipulation of clusters, interrelations of blocks, formulation of propositions, and projection to application cases).

The research presented in this section operationalized a new methodology for theory fusion. The literature investigation conducted revealed that currently available theorizing methodologies can provide limited support in multidisciplinary domains, triggering the need to tailor new approaches. With this need in mind and the inspection of existing methodologies for theory forming, the principles of axiomatization were adopted in the process of giving birth to a new qualitative approach able to combine different (and distant) theories into one multidisciplinary theory. This new methodology is called ATF. As a reflective action, the feasibility, the usability, and the performance of ATF have been validated through an application case.

Having completed the practical application, our main findings were as follows:

- The process of ATF is novel and unique. This fact has been proven by our literature investigation, which did not find any previous publication that applied the same reasoning and approach in merging product-engineering theories.
- The process of ATF involves a level of complexity due to its multiple steps, its cognitive intensity, and its context-driven applications.
- ATF allowed the merging of interdisciplinary and multidisciplinary theories. The starting assumption stated that combined theories deliver more knowledge than individual theories. Based on the results obtained in the application case, we concluded that this assumption is correct. This is because combined theories deliver not only abstract knowledge but also implied sets of functionalities and requirements to be implemented.
- Through ATF, not only can theories be merged, but new theories can be birthed to cover literature gaps in certain contexts. This was the case in the studied application: individual theories did not reflect much about building an SDATB, but when put together they produced a theory about SDATB realization.
- The outcomes of the application case helped to fill in the gaps in design and data analytics theories (multidisciplinary aspect). The obtained propositions form a knowledge platform for the development of an SDATB prototype.
- The knowledge obtained through the application case formed a skeleton of an SDATB by providing a list of requirements and techniques to be used in implementing one. This list will be further studied and filtered to identify the priorities among the elements in the process of conceptualization.
- The ATF methodology may be instrumental in identifying and discarding faulty and weak theories. Correspondingly, it may expose the explanatory and predictive powers of strong and comprehensive theories.
- The demonstrative application revealed that ATF is a relevant methodology for fusing theories for the development of smart computational systems. We conclude that ATF can be used in both interdisciplinary fields and multidisciplinary fields.

3.5. Conclusions

The QBI revealed dissatisfactions that differed from those seen with the actual DATs represented as means in a list of ten needs. These needs were organized and grouped into three clusters: (i) interfacing, (ii) data management, and (iii) reasoning (see Figure 3.20). The major findings were that white goods designers miss (i) advice concerning applicability of data analytics tools, (ii) assistance in using these tools, (iii) support for acquiring and combining data from multiple data sources, (iv) combined and complementary use of qualitative and quantitative input data, and (v) means for fusing the output data of various commercialized tools. The QBI brought attention to the fact that white goods designers less commonly used MoLD to improve functionalities and implementations of their products and that the use of data analytics tools was often limited.

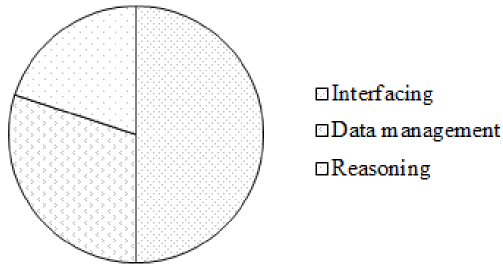


Figure 3.20. Categorization of designers needs

Designers clearly mentioned their inability to use these tools to convert product data into problem-solving knowledge ideally used for product enhancements and the creation of new products. Based on this, the following main propositions were made:

- novel pre-programmed smart interface functions;
- sophisticated data management functions allowing merging of multiple data streams from different sources; and
- artificial intelligence, reasoning mechanisms, awareness, strategy developments, and system adaptation capabilities

were to be considered and included in the next generation SDATB to satisfy white goods designers' practical needs. These functionalities referred to by designers all include a level of smartness, even in behavior, reasoning, and learning. This moves in a direction indicating that a next generation data analytics toolbox should be an SDATB.

Including these proposed affordances within a data analytics toolbox will allow designers to benefit from processing data (particularly MoLD) and to generate knowledge for enhancing white goods. Concerning the application of ATF in the context of SDATB generation, the developed theory was able to deliver more knowledge than individual theories. The outcomes of ATF in this context were formulated as a list of 81 propositions converted into requirements for the SDATB and categorized into clusters: (i) decision-making, (ii) algorithmic concepts, (iii) learning, (iv) data management, (v) interfacing, (vi) reasoning, (vii) data types and characteristics, (viii) design issues, (ix) analytics techniques and methods, and (x) outputs (Figure 3.21).

The propositions obtained reflected the functionalities that need to be included in a next generation SDATB. Some of these functionalities are already provided by existing tools, such as analyzing labeled data and modeling large-scale data. They also reflected what should not be part of the toolbox, such as (i) traditional analytics techniques inadequate for handling big data from smart products and (ii) deep neural networks (DNNs) that are computationally expensive and require long training times for pattern recognition. The major findings of the ATF application were the novel functionalities not yet covered by existing data analytics tools (see Table 3.15). The next chapter discusses how the requirements and functionalities obtained from the previous studies were filtered and enhanced to be used in the conceptualization of a complete SDATB.

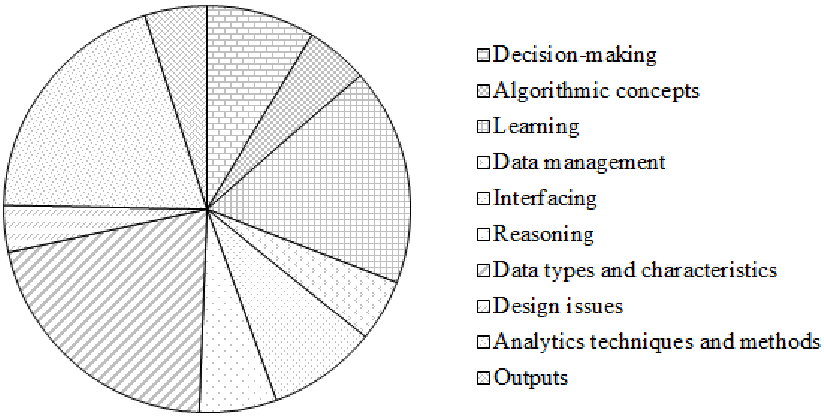


Figure 3.21. Categorization of smart data analytics toolbox requirements

Table 3.15. Major findings of the axiomatic theory fusion application

General function	1st level of sub-function	2nd level of sub-function
Decision-making	SDATB includes context-driven decision-making	SDATB considers dynamically-integrated knowledge
		SDATB anticipates context changes
	SDATB includes proactive decision-making	SDATB takes action proactively
		SDATB allows predictive analytics
		SDATB analyses future opportunities (effective decision-making)
Algorithmic concepts	SDATB algorithms processes complex data	SDATB algorithms deal with high dimensionality and sparseness
Learning	SDATB allows semantic interpretation	SDATB includes ontologies and reasoning engines
		SDATB includes reasoning engines
Data management	SDATB blends data and datasets	SDATB combines all data types
		SDATB combines qualitative and quantitative data
	SDATB merges data streams	SDATB merges data from different sources
	SDATB guarantees storage	SDATB allows high speed storage
Interfacing	SDATB permanently advices designers in their choices	SDATB is permanently accessible
		SDATB recognizes its user
		SDATB helps its user in his choices (step by step)
Reasoning	SDATB allows case-based reasoning	SDATB detects the context of the analysis
		SDATB reasons with cases
		SDATB offers solutions based on saved manipulations
Data types and characteristics	SDATB processes structured, semi-structured and multi-structured data	SDATB combines qualitative and quantitative data
		SDATB merges different data format
Design issues	SDATB proposes solutions to solve difficult design problems	SDATB proposes solutions in the context of the analysis
		SDATB includes predictive analytics
		SDATB analyses future opportunities
Analytics techniques and methods	SDATB predicts future outcomes	SDATB includes predictive analytics
Outputs	SDATB derives actionable insights	SDATB provides outputs in a context

3.6. References

[1]. Sabou, M., Kantorovitch, J., Nikolov, A., Tokmakoff, A., Zhou, X., & Motta, E. (2009). Position paper on realizing smart products: Challenges for semantic web technologies. In: *Proceedings of the 2nd International Conference on Semantic Sensor Networks*, 522, 135-147.

[2]. Porter, M.E., & Heppelmann, J.E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 64-88.

[3]. Almada-Lobo, F. (2016). The Industry 4.0 revolution and the future of manufacturing execution systems. *Journal of Innovation Management*, 3(4), 16-21.

[4]. Prause, M., & Weigand, J. (2016). Industry 4.0 and object-oriented development: incremental and architectural change. *Journal of Technology Management & Innovation*, 11(2), 104-110.

[5]. Eng, J. (2003). Sample size estimation: how many individuals should be studied? *Radiology*, 227(2), 309-313.

[6]. Shokouhi, S.V., Skalle, P., & Aamodt, A. (2014). An overview of case-based reasoning applications in drilling engineering. *Artificial Intelligence Review*, 41(3), 317-329.

[7]. Marling, C., Rissland, E., & Aamodt, A. (2005). Integrations with case-based reasoning. *The Knowledge Engineering Review*, 20(3), 241-245.

[8]. Nguyen, B.V., Burstein, F., Fisher, J., & Wilson, C. (2011). Taxonomy of usage problems for improving user-centric online health information provision. In: *Proceedings of the 16th Americas Conference on Information Systems*. Retrieved from: http://aisel.aisnet.org/amcis2011_submissions/65.

[9]. Agrawal, G., & Su, Y. (2012). Light-weight data management solutions for visualization and dissemination of massive scientific datasets - Position paper. In *2012 SC Companion: High Performance Computing, Networking, Storage and Analysis*, IEEE, 1296-1300.

[10]. Govindugari, D., & McGoveran, D. (2004). Method and architecture for data transformation, normalization, profiling, cleansing and validation. *U.S. Patent Application No. 10/635*,891.

[11]. Descrimes, M., Zouari, Y.B., Wery, M., Legendre, R., Gautheret, D., & Morillon, A. (2015). VING: A software for visualization of deep sequencing signals. *BMC Research Notes*, *8*(1), 419.

[12]. Agrawal, D. (2014). Analytics based decision making. *Journal of Indian Business Research*, *6*(4), 332-340.

[13]. Campain, A., & Yang, Y.H. (2010). Comparison study of microarray meta-analysis methods. *BMC Bioinformatics*, 11(1), 408.

[14]. Deng, X., Su, H., & Zhan, J. (2008). Integration of multiple data sources to simulate the dynamics of land systems. *Sensors*, 8(2), 620-634.

[15]. Navdeep, P., Arora, M., & Sharma, N. (2016). Role of big data analytics in analyzing e-Governance projects. *Gian Jyoti E-Journal*, 6(2).

[16]. Happ, M.B., Dabbs, A.D., Tate, J., Hricik, A., & Erlen, J. (2006). Exemplars of mixed methods data combination and analysis. *Nursing Research*, 55(2), S43-S49.

[17]. Demchenko, Y., De Laat, C., & Membrey, P. (2014). Defining architecture components of the Big Data Ecosystem. In: *Proceedings of the International Conference on Collaboration Technologies and Systems*. IEEE, 104-112.

[18]. Aamodt, A. (2004). Knowledge-intensive case-based reasoning in creek. In *Proceedings of the European Conference on Case-Based Reasoning*, Springer, 1-15.

[19]. Saputelli, L., Verde, A., Haris, Z., & Díaz, D. (2015). Deriving unconventional reservoir predictive models from historic data using case base reasoning (CBR). In: *Proceedings of the Unconventional Resources Technology Conference*, 206-216.

[20]. Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health information Science and Systems*, 2(1), 2-10.

[21]. Wong, P.C., Shen, H.W., Johnson, C.R., Chen, C., & Ross, R.B. (2012). The top 10 challenges in extreme-scale visual analytics. *IEEE Computer Graphics and Applications*, 32(4), 63-67.

[22]. Reed, K.L. (1984). Understanding theory: The first step in learning about research. *The American Journal of Occupational Therapy*, 38(10), 677-682.

[23]. Zhao, F., & Maher, M.L. (1988). Using analogical reasoning to design buildings. *Engineering with Computers*, 4(3), 107-119.

[24]. Gupta, D., & Maheswari, J.U. (2019). Affordance-based Design Method: A Case Study of University Campus. In: *Proceedings of the IOP Conference Series: Earth and Environmental Science*, 323(1), 012158-012164.

[25]. Andrienko, N., & Andrienko, G. (2006). Exploratory analysis of spatial and temporal data: A systemic approach. Springer Science & Business Media.

[26]. Oussous, A., Benjelloun, F.Z., Lahcen, A.A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4), 431-448.

[27]. Chocron, P., Fontaine, P., & Ringeissen, C. (2015). A rewriting approach to the combination of data structures with bridging theories. In: *Proceedings of the International Symposium on Frontiers of Combining Systems*, Springer, Cham, 275-290.

[28]. Burke, C., Macdonald, I., & Stewart, K. (2017). Complementary theories and practices. *Systems Leadership*, Routledge, 289-294.

[29]. Pirner, H.J. (2015). The unknown as an engine of science: A summary. In: *The Unknown as an Engine for Science*, Springer, Cham, 119-123.

[30]. Clegg, S. (2018). Extending the analytical scope of theories of ‘possible selves’. *Possible Selves and Higher Education*, Routledge, 53-68.

[31]. Dhillon, L., & Vaca, S. (2018). Refining theories of change. *Evaluation*, *14*(30), x-x.

[32]. Annapoornima, M.S., & Soh, P.H. (2006). The impact of scientific capability on valuable innovation. In: *Proceedings of the IEEE International Conference on Management of Innovation and Technology*, 1, 335-339.

[33]. Seel, N.M., Lehmann, T., Blumschein, P., & Podolskiy, O.A. (2017). What is instructional design? *Instructional Design for Learning*, Sense Publishers, Rotterdam, 1-17.

[34]. Barnham, C. (2015). Quantitative and qualitative research: Perceptual foundations. *International Journal of Market Research*, *57*(6), 837-854.

[35]. Minton, E.A., Cornwell, T.B., & Kahle, L.R. (2017). A theoretical review of consumer priming: Prospective theory, retrospective theory, and the affective–behavioral–cognitive model. *Journal of Consumer Behaviour*, 16(4), 309-321.

[36]. Locke, E.A. (2007). The case for inductive theory building. *Journal of Management*, 33(6), 867-890.

[37]. Lee, J.S., Pries-Heje, J., & Baskerville, R. (2011). Theorizing in design science research. In: *Proceedings of the International Conference on Design Science Research in Information Systems*, Springer, Berlin, Heidelberg, 1-16.

[38]. Taylor, L.C. (2018). Reassessing and refining theory in qualitative accounting research: An illustrative account of theorizing. *Qualitative Research in Accounting & Management*, 15(4), 510-534.

[39]. Lazar, N. (2013). The big picture: big data computing. *Chance*, 26(2), 28-32.

[40]. Ford, E.D., & Ishii, H. (2001). The method of synthesis in ecology. *Oikos*, 93(1), 153-160.

[41]. Licata, D.R., & Shulman, M. (2016). Adjoint logic with a 2-category of modes. In: *Proceedings of the International Symposium on Logical Foundations of Computer Science*, Springer, Cham, 219-235.

[42]. Bruckmeier, K. (2019). Interdisciplinary knowledge integration for environmental governance: Epistemological questions. *Global Environmental Governance*, Palgrave Macmillan, Cham, 153-178.

[43]. Suddaby, R., Hardy, C., & Huy, Q.N. (2011). Introduction to special topic forum: where are the new theories of organization? *Academy of Management Review*, 36(2), 236-246.

[44]. Rakotonirainy, A. (2005) Design of context-aware systems for vehicle using complex systems paradigms. In: *Proceedings of the Workshop on Safety and Context in Conjunction with CONTEXT’05*, Paris, 43-51.

[45]. Stolt, R., Elgh, F., & Andersson, P. (2017). Design for inspection-evaluating the inspectability of aerospace components in the early stages of design. *Procedia Manufacturing*, 11, 1193-1199.

[46]. Chen, D. (2002). Developing a theory of design through a multidisciplinary approach. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 3, 6-11.

[47]. Hwang, H. (2011). Rational irrationality. Doctoral dissertation, University of New South Wales Sydney.

[48]. Grabowski, A. (2014). Efficient rough set theory merging. *Fundamenta Informaticae*, 135(4), 371-385.

[49]. Oswick, C., Fleming, P., & Hanlon, G. (2011). From borrowing to blending: Rethinking the processes of organizational theory building. *Academy of Management Review*, 36(2), 318-337.

[50]. Hopper, T., & Major, M. (2007). Extending institutional analysis through theoretical triangulation: regulation and activity-based costing in Portuguese telecommunications. *European Accounting Review*, 16(1), 59-97.

[51]. Modell, S. (2015). Theoretical triangulation and pluralism in accounting research: a critical realist critique. *Accounting, Auditing & Accountability Journal*, 28(7), 1138-1150.

[52]. Okhuysen, G., & Bonardi, J.P. (2011). The challenges of building theory by combining lenses. *Academy of Management Review*, 36(1), 6-11.

[53]. Modell, S., Vinnari, E., & Lukka, K. (2017). On the virtues and vices of combining theories: The case of institutional and actor-network theories in accounting research. *Accounting, Organizations and Society*, 60, 62-78.

[54]. Dubois, A., & Gadde, L.-E. (2002). Systematic combining: An abductive approach to case research. *Journal of Business Research*, 55, 553-560.

[55]. Grabowski, A. (2016). Tarski's geometry modelled in Mizar computerized proof assistant. In: Proceedings of the *Federated Conference on Computer Science and Information Systems*, 373-381.

[56]. Hayashi, Y., Bourdeau, J., & Mizoguchi, R. (2006). Ontological modeling approach to blending theories for instructional and learning design. *Frontiers in Artificial Intelligence and Applications*, 151, 37-44.

[57]. Hayashi, Y., Bourdeau, J., & Mizoguchi, R. (2006). Ontological support for a theory-eclectic approach to instructional and learning design. In: *Proceedings of the European Conference on Technology Enhanced Learning*, Springer, Berlin, Heidelberg, 155-169.

[58]. Mayer, K.J., & Sparrowe, R.T. (2013). Integrating theories in AMJ articles. *Academy of Management Journal*, 4, 917-922.

[59]. Weisstein, E. (2004). Axiomatic system. Retrieved from: Mathworld: http://mathworld.wolfram.com/AxiomaticSystem.html.

[60]. Turner, J.H. (1987). Analytical theorizing. *Social Theory Today*, 156-194.

[61]. Bailey, K.D. (1970). Evaluating axiomatic theories. *Sociological Methodology*, 2, 48-71.

[62]. Guichun, G. (2010). The boundaries of context and their significance. *Frontiers of Philosophy in China*, 5(3), 449-460.

[63]. Rodin, A. (2014). Axiomatic method and category theory. Springer, Berlin.

[64]. Bandodkar, N. (2015). Valuable or stagnating? An essay on axiomatic theories in IS research. In: *Proceedings of the 21st Americas Conference on Information Systems*, 5, 3558-3566.

[65]. Zeng, Y. (2002). Axiomatic theory of design modeling. *Journal of Integrated Design and Process Science*, 6(3), 1-28.

[66]. Strasnick, S. (1979). Moral structures and axiomatic theory. *Theory and Decision*, 11(2), 195-206.

[67]. Cellucci, C. (2019). Diagrams in mathematics. *Foundations of Science*, 1-22.

[68]. Gillies, D. (2012). *Philosophical theories of probability*. Routledge, London.

[69]. Pogliani, L., & Berberan-Santos, M.N. (2000). Constantin Carathéodory and the axiomatic thermodynamics. *Journal of Mathematical Chemistry*, 28(1-3), 313-324.

[70]. Sayin, U. (2015). A short introduction to system theory: Indispensible postulate systems and basic structures of the systems in quantum physics, biology and neuroscience. *NeuroQuantology*, 14(1), 126-142.

[71]. Quinn, F. (2011). Contributions to a science of contemporary mathematics. *Science of Mathematics*. Retrieved from: http://www.math.vt.edu/people/quinn/history_nature/nature0.pdf.

[72]. Halbach, V. (2014). Axiomatic theories of truth. Cambridge University Press.

[73]. Qui, J., Zhang, J., Ma, Y., Feng, J., & Wang, P. (2018). Reliability-based robust design for kinematic accuracy of a mechanism with axiomatic theory. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 232(13), 2412-2423.

[74]. Gioia, D.A., & Pitre, E. (1990). Multiparadigm perspectives on theory building. *Academy of Management Review*, 15(4), 584-602.

[75]. Aban-Perez, R. (2018). Gerontology embracing technology. *International Journal of Current Research*, 10(8), 72479-72483.

[76]. Kainrath, F. (1999). A divisor theoretic approach towards the arithmetic of Noetherian domains. *Archiv der Mathematik*, 73(5), 347-354.

[77]. Suppe, F. (2017). Axiomatization. *A Companion to the Philosophy of Science*, 9-11.

[78]. Morgan, T.R., Tokman, M., Richey, R.G., & Defee, C. (2018). Resource commitment and sustainability: a reverse logistics performance process model.

International Journal of Physical Distribution & Logistics Management, 48(2), 164-182.

[79]. Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information systems,* 8(5), 312-335.

[80]. Cash, P., & Piirainen, K.A. (2015). Building a cohesive body of design knowledge: developments from a design science research perspective. In: *Proceedings of the 20th International Conference on Engineering Design*, 10, 241-250.

[81]. Colander, D. (2005). The future of economics: The appropriately educated in pursuit of the knowable. *Cambridge Journal of Economics*, 29(6), 927-941.

[82]. Horváth, I. (2019). Combining unmergeables: A methodological framework for axiomatic fusion of qualitative design theories. In: *Proceedings of the* International Conference on Engineering Design, Design Society, 3591-3600.

[83]. Adalı, S. (2013). Modeling trust context in networks. Springer, Berlin.

[84]. Zhang, Z., & Zha, H. (2006). A domain decomposition method for fast manifold learning. *Advances in Neural Information Processing Systems*, 1625-1632.

[85]. Gilbert, J.R., Moler, C., & Schreiber, R. (1992). Sparse matrices in MATLAB: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1), 333-356.

[86]. Abou Eddahab, F.-Z., & Horváth, I. (2018). What do designers miss regarding the outputs of data analytics tools in the context of possible product improvements? In: *Proceedings of the 12th Symposium on Tools and Methods of Competitive Engineering*. 1-14.

[87]. Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2018). Big data analytics: Computational intelligence techniques and application areas. *Technological Forecasting and Social Change*. Retrieved from: https://doi.org/10.1016/j.techfore.2018.03.024.

[88]. Sruthika, S., & Tajunisha, N. (2015). A study on evolution of data analytics to big data analytics and its research scope. In: *Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems*, 1-6.

[89]. Mishra, M., Sidoti, D., Avvari, G.V., Mannaru, P., Ayala, D.F., Pattipati, K.R., & Kleinman, D.L. (2017). A context-driven framework for proactive decision support with applications. *IEEE Access*, 5, 12475-12495.

[90]. Kim, S.H. (1991). A formal model of creative decision making. *Robotics and Computer-Integrated Manufacturing,* 8(1), 53-65.

[91]. Horváth, I., & Wang, J. (2015). Towards a comprehensive theory of multi-aspect interaction with cyber physical systems. In: *Proceedings of the International Design Engineering Technical Conference*, American Society of Mechanical Engineers, V01BT02A009-V01BT02A009.

Chapter 4

Research cycle 3: Conceptualization of a demonstrative smart data analytics toolbox

4.1. Introduction

4.1.1. Objectives and assumptions of the third research cycle

First, it must be noted that the term "function" is used in this dissertation to depict an activity or computational operation that is the purpose (raison d'être) of the SDATB, while the term "functionality" is used to describe the range of functions that are provided by the SDATB as well as the quality of being suited to provide them for the purpose. In the third research cycle, as a first action, the outcomes presented in the previous chapter were checked with regard to the related literature to verify the actual existence of the uncovered knowledge gaps. This action helped (i) to consolidate the theoretical fundamentals, (ii) to determine some original functionality for the smart toolbox, and (iii) to specify the intended support functions and algorithmic operations. It was also useful (i) to cast light on the issues related to existing data analytics tools from the perspective of designers, (ii) to capture the domain of possible improvements in the age of smart tools and products, and (iii) to identify the opportunities for the amelioration of designers' experience using DATs.

In addition to what is mentioned above, it has been revealed what range of new functions a next generation SDATB needs to (and can) include. To impose a logical order, the following sets of functions have been identified: (i) basic functions (uniquely novel and specialized computational functions such as learning and reasoning functions), (ii) auxiliary functions (common or dedicated computational processing and management functions such as those related to data management), and (iii) interface functions (human and/or system interaction and communication functions such as data input and output visualization functions). This logical scheme rationalized the thinking about the functional conceptualization and the detailed elaboration of the functions of the foreseen SDATB. The research work and its results are also presented according to this scheme in this chapter.

The abovementioned categories of toolbox functions were specified and elucidated considering an overall vision, but given the concomitant complexity and the amount of nonscientific research work, a representative subset has been considered for full-fledged elaboration. Based on the prioritization of functions in each category, we have completed a demonstrative realization of some representative functions as well as significant basic, auxiliary, and interface functions of the SDATB. As a step towards the targeted computational realization, we detailed the chosen main smart functions and decomposed them into low-level functions. This decomposition was used in architecting the computational modules by which the main functions could be operationalized.

4.1.2. Methodology applied in the third research cycle

As mentioned above, the third research cycle focused on the ideation and the technical conceptualization of a demonstrative SDATB. Consequently, research cycle 3 was framed according to the DIR methodology. We organized research activities of this cycle in three phases: (i) explorative, (ii) constructive, and (iii) confirmative. The explorative phase concentrated on examining knowledge and enabling technologies for the toolbox conceptualization. The entry point of this phase was the synthesized findings of activities from the second research cycle. At the end of this phase, fundamental concepts related to the SDATB were specified. The constructive phase focused on the establishment of a comprehensive conceptual model of a demonstrative SDATB. Accordingly, we investigated the functions of the toolbox, articulated the concept, completed the functional decomposition, and built a high-level architecture. The confirmative phase focused on testing the feasibility of all computational constructs of the SDATB. In this sense, a plan for a software prototype implementation was developed.

4.2. Fundamentals of foreseen smart data analytics toolbox

4.2.1. Literature investigation

Carrying out a literature study to concretize the ideation of the toolbox was an essential step in conducting this research. The objective was to further investigate the findings of previous studies in the literature and determine to what extent possible matching can occur. In addition, we checked the obtained requirements to study their feasibility. The starting point was the two complementary theories: (i) the need theory from the web-based interrogation and (ii) the theory built by ATF. As observed from their formulation, the next generation data analytics toolbox should be smart. Accordingly, the smartness of data analytics tools and their possible affordances were the focus of the literature study. By referring to the clustering presented in the previous chapter, we identified identical clusters in both studies – (i) interfacing, (ii) data management, and (iii) reasoning – which implied their deep investigation in the literature search.

Moreover, by referring to the outcomes of the ATF application, some clusters emerged as predominant in terms of the toolbox requirements they contained, mainly (i) learning, (ii) data types and characteristics, and (iii) analytics techniques and methods. These clusters also required particular attention in the investigation into the state of the art. The following statement led us to the reasoning model of the literature study presented in Figure 4.1. The reasoning model does not include “data types and characteristics” due to the overlap between the components of that cluster and the components of “data management,” such as dataset merging, where the type of data need to be investigated. For this reason, only data management was kept. In addition, data analytics techniques

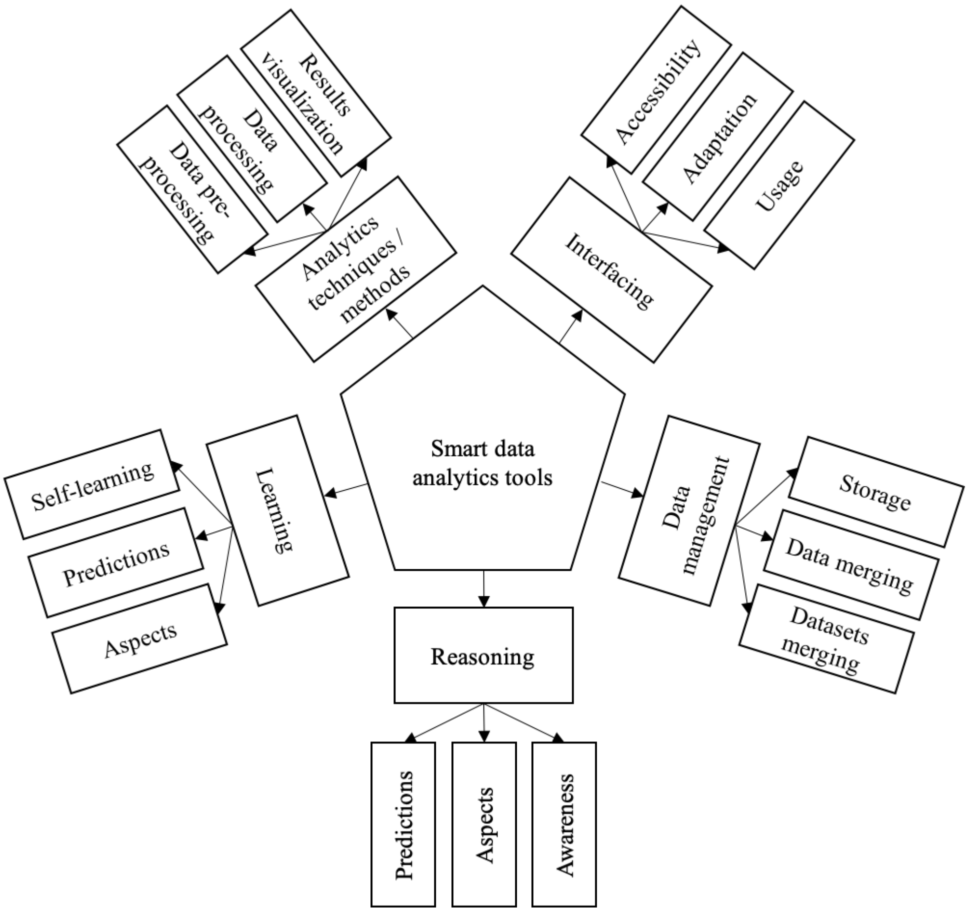


Figure 4.1. Reasoning model of the literature investigation

and methods had been discussed in the first and second research cycles. The focus in this cycle was on predictions using data analytics techniques and methods. Given its overlap with the other components on the graph, predictions were sub-parts of the remaining components.

As can be seen in the reasoning model, smartness was not included explicitly in the aspects of the literature study because the context of the study was smart data analytics. The research included the investigation of the SDATB in terms of the following:

- the provided level of accessibility, its adaptation to the needs of a particular user, and the way it could be used to support and help the user in manipulations of the tool or toolbox;
- data management within an SDATB in terms of allowed storage and the possibility of merging different data types from different sources;
- reasoning within an SDATB in terms of its awareness of the context of the study and the situation in which it was used, the affordances regarding reasoning aspects, and the possibility of making predictions based on reasoning with data; and
- learning happening within an SDATB, its nature (self-learning or not), the affordances in various aspects of learning, and making predictions based on learned analyses and procedures.

For the mentioned concerns, the data were collected from scientific publications and professional documents, respectively, via web-search engines (Google Scholar, Web of

Science, Research Gate, and so on) using the component phrases from the reasoning model, their synonyms (smart data mining was added to smart data processing), and their detailing as keywords. Key terms that were more specific were extracted from the explicit theories of the two previous studies.

Smart data processing is supposed to be able to retrieve data efficiently [1]. As reported, smart data analytics tools are rarely used in design and manufacturing, making manufacturing systems unable to take advantage of vast data amounts generated in industry [2]. Big data cannot be managed in this way [3]. It has been observed that the smartness of data analytics tools is linked to machine learning [4] or deep learning [5]. These approaches present different characteristics. Machine learning offers computational and analytical solutions in integrative analysis of sufficiently large heterogeneous datasets [6]. It is mostly designed for labeled data loaded in full into memory, which does not apply in the context of big data [7]. Unfortunately, currently popular approaches do not presume data distribution [8].

Machine learning presents a level of complexity in terms of the amount of data to deal with and its long computation time [9]. Deep learning goes beyond machine learning with the capability of handling complex nonlinear relationships in data [10]. Its architectures have the ability to generate learning patterns and relationships other than those among close neighbors in data [11]. It can be used for labeled data, but it is mainly attractive for its ability to learn from large amounts of unlabeled, unstructured data and extract meaningful representations and patterns from big data [12]. However, (i) it does not provide formal proofs to ensure high-quality performance, (ii) it lacks mechanisms for learning abstractions from explicit data, (iii) its extracted patterns are more superficial than they appear at first, and (iv) it has not been well integrated with prior knowledge [13]. Because of the diversity of deep learning software tools, it is sometimes difficult to select the most appropriate platform to carry out deep learning tasks [14].

Based on the established reasoning model, we recognized a need to include context in data analytics tools. Consideration of the context may start from the use of the interface and may continue, down the road, to data visualization. In interfacing, a context-sensitive help function can provide information to the user based on the operations performed in software applications. An important point is that the user does not necessarily need to request the help. This differs from traditional help functions, which require the user to look for a specific help topic or select a help topic from a list [15]. In human–computer interface, multimodal interfaces appeared as a trend for building interfaces that are intelligent enough to incorporate users' intuitions and that load actively [16]. They are defined as user interfaces capable of receiving diverse high data and producing diverse outputs in response to it [17]. Such interfaces should adapt to the needs and capabilities of different users as well as to the context of the use. They are characterized by being dynamically adaptive, which enables them to adapt to a change of tasks or contexts. Unfortunately, even with their good opportunities, multimodal interfaces need further research to determine the most effective and intuitive combinations of inputs and outputs for users, applications, and contexts as well as techniques for error handling and adaptive processing [18].

Intelligent user interfaces have capabilities that are more closely associated with humans than with computers in terms of how to (i) perceive, (ii) interpret, (iii) learn, (iv) use language, (v) reason, (vi) plan, and (vii) decide [19]. They guarantee that data are captured in semantically annotated form. This can be realized by determining the underlying context of the user input and expressing it with corresponding semantic terminology [20]. With the rapid growth of intelligent systems, decision-making has become an important research topic in human–computer interface [21]. Designers are looking to integrate their big data and advanced analytics, also called intelligent or smart analytics, into operations to become more analytics-driven in their decision-making [22]. From this perspective, machine learning and other predictions work well in practical scenarios. The popularity of these approaches has created an increasing demand for similar tools that nonexperts can easily use [23].

Big data management is the process of creating value using big data. It includes data collection, cleaning, anonymization, and publishing [24]. In this sense, intelligent (smart) data management is needed for value creation [25]. In dealing with this aspect and managing data efficiently, three main factors are to be considered: (i) the diversity of data, (ii) a plethora of different formats, and (iii) a huge amount of generated data containing noise [26]. In addition, data management needs to handle both historical and streaming data. This poses the challenge of merging these two sources for further analytics, as well as the challenge of providing high-capacity storage [27]. Due to its flexibility, the cloud environment is recommended by many researchers to solve the challenge of storing big data in a smart environment [24]. Furthermore, multiple data management processes are possible in the cloud such as (i) data storage; (ii) data administration; (iii) data access, concealment, and security; (iv) protection of data from unauthorized access; and (v) data sharing [28].

In the context of fusion in smart data management, as discussed in recent publications, it has been recognized that advanced data fusion implies the involvement of artificial intelligence systems such as neural networks [29]. In other words, merging data streams with the help of neural networks is an opportunity to increase the accuracy of merging systems [30]. Merging different data sources allows data enrichment by providing more information about an event than the that communicated by one source [31]. Typically, dataset specific and application programming interface-based plug-ins can be used for combining multiple datasets [32]. Some proposed frameworks for advanced data fusion contain three main steps: (i) data preprocessing (to clean and purify raw data), (ii) processing by a neural network (using preprocessed data as input to train neurons with weights and then leveraging them for further processing), and (iii) use of a fuzzy inference system (to map given inputs to an output using fuzzy set theory) [33]. Today, combining datasets is still a challenge because there is no guarantee of performing merging adequately without compromising relevant information from each dataset [34].

In the literature, reasoning mechanisms are divided into two main categories: (i) inductive reasoning and (ii) deductive reasoning mechanisms [35]. The first category includes analogy-based reasoning, CBR, and probability-based reasoning. The second category includes rule-based reasoning and pattern-based reasoning. One focused investigation of smart reasoning explored that it was exposed as context-dependent reasoning or context-driven reasoning [36]. From this perspective, smart reasoning is implemented by CBR systems [37]. These systems reason from examples, called cases, following four steps: (i) retrieve, (ii) reuse, (iii) revise, and (iv) re-train [38]. In CBR, when a case is presented, those most similar to it are retrieved to make predictions. This is done by matching the features of the given case with the features of other existing cases using a *k*-nearest neighbors (k-NN) algorithm. CBR is characterized by transparency, since it reasons from precedent examples, similar to what humans often do [38]. Recently, CBR has been combined with artificial neural networks (ANNs) to form a twin system to meet system requirements of accuracy and interpretability. The ANN is used in the k-NN retrieval step of CBR to identify the nearest-neighbor case and explain the ANN prediction [39].

A more recent augmentation is the association of DNNs with CBR as twins. Other researchers built a CBR mechanism into a DNN architecture to avoid the need for post hoc explanations (as to why artificial intelligence produced the outputs it did) [40]. DNNs have been explained using simple proxy systems such as (i) linear models, (ii) decision trees, (iii) automatic rule extraction, and (iv) saliency mapping [41]. Memory-based reasoning (MBR) is also used for prediction purposes following CBR principles. MBR reasons from "past experiences," also called cases. The difference is that it does not build a model from training data. For prediction or classification, MBR builds a specific local model by finding k-NNs of the test sample, and it combines the pieces of information by averaging or voting [42].

Our first finding related to smart learning was that the concept of smart learning constituents is still in its infancy [43]. Consequently, there is a need for effective methods to improve assessment processes in the context of smart learning environments [44]. A smart learning environment is defined as "a learning place or activity space that (i) can perceive learning scenarios, (ii) identify the characteristics of learners, (iii) provide appropriate learning resources and convenient interactive tools, (iv) automatically record the learning process, and (v) evaluate learning results, so as to promote learners' effective learning" [45]. Thus far, ANNs are regarded as the best way to embed learning and intelligence into digital devices, but they require training and large amounts of heavy floating-point calculations [46]. As discussed in the literature, the new trend of learning, the so-called smart learning, is simply self-learning. This means that decision-making models within systems or software tools are able to learn from big data to improve themselves. This is done by embedding deep machine learning (DML) into decision-making models. The other side of the coin is that application systems should be equipped with continuous learning capability. These models will be extended and fragmented into new models. This way, smart decision-making can be achieved by (i) picking up precise data as parameters, (ii) determining resolutions quickly, and (iii) evaluating results sufficiently [47].

4.2.2. Requirements for a smart data analytics toolbox

Traditional data analytics tools cannot be applied directly to big data [48] or to managing big data to extract practical knowledge from them [49]. This points to the need for novel and sophisticated data analytics tools. In a broader sense, what is actually needed is to add smartness and learning capabilities to a wide range of computer systems (including data analytics tools and toolboxes) [46]. To cope with the knowledge gap found in the literature concerning smart data analytics tools and toolboxes, we integrated the findings of the QBI and the synthetic theory devised using the ATF methodology. This integral body of knowledge was used to formulate operational requirements for a next generation data analytics toolbox.

On the one hand, the QBI revealed the need for (i) step-by-step assistance, (ii) advice in selecting means, (iii) multifold data visualization, (iv) multichannel data management, (v) blending datasets, (vi) combining qualitative and quantitative data, (v) permanent accessibility, (vi) adaptation to users, (vii) CBR, and (viii) learning from applications. On the other hand, the devised synthetic theory was more concrete about the needs and the requirements for the SDATB. This theory suggested that the SDATB should include (i) context-driven decision-making, (ii) proactive decision-making, and (iii) algorithms able to process complex data. In addition, it should (iv) allow semantic interpretation; (v) blend data and datasets; (vi) merge multiple data streams; (vii) allow high-speed and high-volume storage; (viii) provide permanent accessibility; (ix) deliver advice to designers based on their work context; (x) allow CBR; (xi) process structured, semi-structured, and multi-structured data; (xii) propose solutions to solve difficult design problems; (xiii) predict future outcomes; and (xiv) derive actionable insights.

A deeper investigation of the literature findings revealed that the reported research projects mainly concentrated on handling big data and addressed the concomitant challenges. More specifically they focused on data capturing, curation, and storage and the performance of data processing [50]. The main technical challenges identified were (i) interoperability issues, (ii) usability and programming, and (iii) using big data analytics frameworks [51]. Concerning data analytics tools, the studied publications (i) compared and ranked them [52], (ii) provided detailed descriptions of them [53], and (iii) presented the opportunities they offered [54]. No publication discussed results of a project in which product designers had used smart data analytics tools in their specific design tasks. It is widely promoted that data analytics processes are crucial for product designers. The fact is they usually miss knowledge even about the available variety of data analytics tools and principles for selecting them.

From a computational point of view, some general requirements must be considered to frame and orient the elaboration of the SDATB. These requirements are as follows:

- **General requirements (GR) for the SDATB:**

  GR_1: The computational functions of the SDATB should be robust and efficient in the application context of data analytics tools.

  GR_2: Some functions of the SDATB should be able to interpret (understand) inputs given by the designer.

  GR_3: The computational mechanisms used for realizing the different functions should be linearly computable.

  GR_4: The interactions between the designer and the SDATB must not be error prone.

Based on the further elaboration of the findings of the above-mentioned studies, we distinguished three groups of requirements: (i) basic requirements (BRs), (ii) auxiliary requirements (ARs), and (iii) interface requirements (IRs). We may claim that the ultimate source of the concrete requirements was the synthetic theory derived using the ATF methodology, since it included the outcomes of the QBI (in the form of the so-called need theory). This was useful, since the literature study did not provide any concrete technical requirements in addition to the limited number of general requirements.

4.2.2.1. Basic requirements

BRs are the causes and triggers of the unique and novel basic functions of the SDATB. If the designed functionality of the SDATB did not satisfy these requirements, the toolbox would not qualify as a smart and knowledge-enabled recommendation system, which is implied by its name. Consequently, we can say that fulfilment of the BRs is a necessary and sufficient condition for the implementation of the smart toolbox. However, the BRs primarily determine only the expected smart operation, manifestation, and behavior. For this reason, they represent only a subset of the total set of operational requirements. They do not encompass those requirements, which are related to computational managing and handling of big data or to interaction and communication with the entities of the external environment or within the smart toolbox.

The following BRs have been formulated:

BR_1: The SDATB should be able to fuse MoLD streams

Processing multiple concurrent data streams is an obvious task for the SDATB. By including a multisource data fusion technology, an SDATB can (i) eliminate redundant and contradictory data obtained from various sources, (ii) reduce the uncertainty of provided information, (iii) develop a nearly complete description of the monitoring environment, and (iv) enhance the accuracy of decision-making by intelligent systems [55]. It can also build better situational awareness and reasoning capabilities, as well as reduce its response time [56]. During recent decades, data fusion has evolved rapidly in various application fields [57] [58]. Data fusion is a synthesis of incomplete information about environmental features provided by multiple data sources. The goal is to establish a relatively consistent and complete description through a more complete and accurate set of information [55].

BR_2: The SDATB should recognize patterns within a context

Context is the informational neighborhood surrounding a point of interest [59]. It affects the realization of certain tasks in which the needed knowledge depends on specific data, a specific user, or a particular environment. Contextual information has an important role in many pattern-recognition tasks [60]. In the case of data analytics for design improvement, the context, the objectives, and the inputs of analysis differ from one designer to another. The type of data collected (use data, maintenance data, etc.), the kind of product to be enhances, and the expected results differ from one case to another depending on the context. As variables, these influence the possible objective and the overall approach of pattern recognition. This influence underlines the importance of integrating context information in data analytics systems. Developers of context-aware systems have defined context information as all information related to people, places, or objects relevant for the operations of systems [61]. A context-aware SDATB will improve its performance time-to-time [62] and may evolve towards a system that mimics the capacities of human beings in terms of offering adaptive analyses and decision-making in changing circumstances [63].

BR$_3$: The SDATB should be able to build situation awareness

Situation awareness is defined as “knowing what’s going on so you can figure out what to do” [64]. More specifically, it is “the perception of elements in the environment within a volume of space and time, the comprehension of their meaning, and the projection of their status in the near future” [65]. Situation awareness manages the continuous extraction of environmental information and integrates it with prior knowledge, with a view to further direct perception and anticipation of upcoming events [66]. Starting from the finding that situation-aware systems gather, process, and interpret large amounts of data, we propose that equipping the SDATB with a situation awareness capability enhances the context-dependent performance of the system [67]. Such a capability helps designers gain a better overview of what is happening and, consequently, helps them make effective decisions and take appropriate actions [68].

BR$_4$: The SDATB should be able to reason with past cases and learn from applications

Solving problems by recalling and learning from previous similar cases is called CBR. This can enhance the task-related recommendation services of the SDATB. The stored (and annotated) cases are used as templates to solve problems relying on a characteristic overlap [69]. They are stored in a general knowledge repository, and the computational mechanism provides the retrieval and reasoning functions [63]. In the case of SDATB, a “pattern recognition–conflict resolution–action” cycle can be repeated until an adequate solution is obtained or until no additional applicable cases are found in the case base [70]. Consequently, former experiences will be used to predict and activate similar cases or will be adapted to new solutions [70]. Inclusion of CBR in the SDATB has three main advantages: (i) remembering previous situations similar to the current one and using them to solve a new problem, (ii) understanding a new problem in terms of previous experiences, and (iii) adjusting an old solution to meet demands of a new issue [71].

4.2.2.2. Auxiliary requirements

ARs imply the operational functions that are closely associated with the fulfilment of the basic functions. They are needed for completing the computational operations, but they may also imply smart operations. The range of the ARs depends on the basic functions, but they can be applied to the whole of the SDATB so as privacy, security, and dependability. It must be noted that ARs complement and articulate the BRs by formulating the need for management, transformation, warehousing, and operation scheduling of data, knowledge, tools, and other enablers.

The most important ARs have been formulated below:

AR_1: The SDATB should provide recommendations for tools selection

In general, recommendations have an important role in all life areas [72] [73]. A recommender system is a rating (a preference of priority evaluation) system for a product or a service. A recommender system is a quasi-intelligent system that suggests items of interest to the user. These systems can even be found in YouTube and Facebook. The type of recommender system depends on the domain and the purpose of the application. In general, a recommender system is viewed as a subclass of information filtering systems used to predict user preferences. The content analysis of the needed or possible recommendation can be done using two main methods: (i) content-based analysis or (ii) collaborative analysis [74]. Recommender systems use various information, knowledge, and pattern-mining techniques and semantic content generation technologies [75] [76]. By providing examples of such techniques and computational technologies for specific applications, the literature demonstrates that there is a link between recommendation systems and data analytics. In fact, the kernels of recommender systems are built using probabilistic data analytics and processing techniques, but they also use artificial narrow intelligence technologies. Such systems are invading most online marketing and other utility systems. Despite this, current data analytics systems are yet not augmented with recommender systems. As indicated by the interviewed practical designers, a recommender system is supposed to help designers select the most appropriate tool for a task, or recommend data processing tools to designers based on what they have learned about their previous applications and performances.

AR_2: The SDATB should be individually customizable

A customized SDATB is a system that satisfies users' individual requirements concerning the operation, appearance, and experiences [77]. This means that the SDATB should be designed and implemented according to the particular needs of the users [78]. For instance, it can automatically set up the preferred working environment for a particular designer based on his or her learned choices and preferences. In this way, the SDATB may prevent the designer's getting lost in the overwhelming number of functional possibilities and processing options offered by the current integrated toolboxes. Technically this means that the functional options and use features of a smart toolbox should be not fixed and extendable. It should be possible to add a service when it is needed by the designer, and to hide, rather than erase, a service when it is not needed or preferred by the designer.

AR_3: The SDATB should be able to merge quantitative and qualitative MoLD

Qualitative data is defined as "empirical information about the world, not in the form of numbers. Most of the time (but not always …) this means words." Quantitative data is "empirical information in the form of numbers, produced by measurements" [79]. Qualitative output data are results of interpretation, while quantitative output data are results of calculation. By merging them, a more complete view can be presented and richer analyses of MoLD can be facilitated. However, combining qualitative and quantitative types of data raises a methodological challenge. The challenge is associated with the understanding of how to merge the two data types purposefully and smartly in the analysis process to obtain coherent, reliable, and valid results [80]. This is an important issue because, as argued in the literature, merging quantitative and qualitative MoLD would allow the SDATB to perform a deeper study of a phenomenon than can be done when the complementary qualitative and quantitative datasets are used separately [81].

***AR$_4$*: The SDATB should provide a dynamic and high-volume storage capacity**

Data storage is the process of organizing, warehousing, and managing massive data. Typically, a storage system is divided into two layers that are formed by the storage infrastructure and the data management software. The first layer includes all of the storage devices and the network devices connecting them. The second layer is populated by the computational mechanisms (software agents) that are important for the realization of a scalable, effective, and reliable storage system to support real-time big data analytics. In practice, it contains (i) the file system, (ii) the database management system, and (iii) the distributed computing [82]. If the SDATB provides high-volume, dynamic storage capabilities, it will solve three main challenges of data storage: (i) reliability and persistency of data storage by including long-term and short-term storage while balancing the cost caused by the tremendous amount of MoLD [83], (ii) scalability by taking into account the volume and heterogeneous characteristics of MoLD, and (iii) efficiency by supporting the vast number of concurrent access queries from the data analytics phases [82].

***AR$_5$*: The SDATB should provide permanent online accessibility**

Permanent accessibility of devices is facilitated by online technologies, which are defined as “a diverse set of technological tools and resources used to communicate, and to create, disseminate, store, and manage information” [84]. An SDATB connected to or residing on the Internet will offer the needed permanent online accessibility and will allow a continuous exchange of information [85]. In addition, working with massive data streams requires multiple channels and broad bandwidth. If these are guaranteed, designers can analyze their MoLD at any time and in any location.

4.2.2.3. Interface requirements

IRs play an important role in fully interactive or quasi-interactive systems, as opposed to fully or partially automated systems. IRs concern the interaction and communication with human stakeholders (e.g. end users and knowledge engineers) but also the internal interoperation among the system components and the external interoperation and communication with other systems. In the case of the SDATB, the interface functions that connect the system to the user are important for both interaction (control) and communication (informing). The most important IRs ensure visualizations, connections, communication, and representations.

The following IRs have been specified:

IR_1: The SDATB should provide trustful authentication and identification

User authentication is an important factor in controlling unauthorized access to systems [86], and it can also support setting up the working environment based on the user's identity and the state of work. The issue of trustful authentication and identification obtained international attention related to smart environments [87]. User authentication is the process by which a system verifies whether the user has a legitimate claim to access to the system [88]. There are three main approaches used for authentication: (i) possessions-based authentication (uses a unique physical item such as passport, smartcard, or key), (ii) knowledge-based authentication (uses secret information such as password), and (iii) biometric-based authentication (measures a unique human characteristic or trait such as a user's face, DNA, ear shape, signature, or voice) [89]. Artifacts for the first type of authentication can be shared, duplicated, stolen, or lost [90]. Means for the second type, even though the type is widely used, are not without problems. Many passwords are easy to guess, and they can be shared with others or forgotten [91]. Consequently, the SDATB should include biometric-based authentication, because biometric features are in general difficult to reproduce, and they cannot be lost or forgotten [92]. This will allow a secure processing environment for the designer.

IR_2: The SDATB should provide help for system-level navigation for designers

The availability of "help" service is one of the criteria to measure the suitability of data analytics tools [93], and it will be necessary in next generation smart systems too. Having a content- and context-sensitive comprehensive help function in the SDATB will assist the designer during all sessions of toolbox use. While most software tools have topic-oriented help functions, the help function of the SDATB should offer smart system-level navigation for designers. This function can be combined with historical learning and/or can be trained by answering questions and by responding to unfamiliar concepts asked about by the designer [94]. This makes the use of the SDATB easier and more efficient.

IR_3: The SDATB should perform continuous process monitoring

Process monitoring is a functional capability linked to the operation and control of complex industrial processes. It targets improvements in a wide range of applications based on monitoring objectives, regularity requirements, and the design of the process facility [95]. One of the usual goals of process monitoring is detection of errors that may lead to a process failure [96]. The evolution and complexity of current data, products, processing tools, and so on has various consequences for the approaches to process monitoring. Traditional approaches of process monitoring are recognized as being no longer efficient because of dimensionality issues [97]. Consequently, novel models are needed to deal with the high dimensionality issues in computational process monitoring, which are caused by the proliferation of the use of sensors. Discrete and networked sensors typically produce a large amount of data on a continuous basis [98]. A sophisticated process monitoring function included in the SDATB will continuously (i) record the actions of the designers and the results of those actions, (ii) detect designer mishaps and errors throughout the entire process, (iii) resolve or eliminate those errors, and (iv) analyze the processes and learn from them, for the benefit of the designers.

IR_4: The SDATB should include a variety of dynamic visualization options

MoLD are distinguished by their variety and changing nature. This is evident if we consider that MoLD may include (i) failure data, (ii) maintenance data, (iii) product age data, (iv) operating environment data, (v) usage intensity data, (vi) maintenance report data, (vii) refund and replacement data, and more [99]. The abovementioned characteristics largely influence the types of visualizations that need to be used with these data [100]. MoLD, like other big data, require flexible and dynamic interactive visualization techniques [101]. For this reason, the SDATB should be equipped with visualization functions that can be selected and applied in line with the characteristics of the processed MoLD. This approach to visualization is becoming recognized as an integral quality enabler in innovation support [102]. Clearly, dynamic visualization is regarded as more vivid than static visualization in representing the variations and trends in MoLD [103].

Many more requirements were formulated in the process of operationalizing the results of the ATF, but we considered only the nontraditional ones (i.e. those requiring a given level of smartness).

4.2.3. Towards the representative functions of the smart data analytics toolbox

The literature casts light on many efforts to investigate and support (i) processing performance, (ii) big data handling, and (iii) data storage of software tools. That is the reason (novelty) why we have chosen to deal with only those functionalities of the SDATB that have not yet been addressed in the literature (i.e. no specific underlying theories or concrete computational solutions have been proposed). Thus we focused on only those requirements that imply smart data management functions and the need for smart computational operations. These were considered to be the basis of the implementation of representative basic, auxiliary, and interface functions of the demonstrative concept of the SDATB. As a secondary factor, we considered the utility of these functions in the context of the design application of the SDATB and of the usability of the SDATB by designers. Among the requirements presented in Section 4.2.2, (i) BR_3 and IR_4 are related to processing performance, (ii) AR_3 refers to big data handling, and (iii) AR_4 refers to storage. Thus, these requirements are not considered in our implementation. The novel requirements, on which we focused our attention, were BR_1, BR_2, BR_4, AR_1, AR_2, AR_5, IR_1, IR_2, and IR_3.

Requirement BR_1 implies a function that fuses middle-of-life data streams (MoLD-Ss). Actually, this function should not only computationally fuse MoLD streams but should also provide recommendations to the designer on what to do with the to-be-merged and merged data streams and how to do it. Therefore, the smart basic function (F_{SB}) implied by BR_1 is named *F_{SB1}: Recommendation for merging MoLD-Ss*. Requirement BR_2 implies a function that belongs to processing performance functions but was not covered in the literature. This requirement can be fulfilled by a function that applies pattern recognition in a given context and performs semantic interpretation of data transformation outcomes. Its corresponding function is *F_{SB2}: Semantic interpretation*. Requirement BR_4 reflects the need for the toolbox to have reasoning and learning capabilities. It must be able to reason from past usage and learn from future applications. The corresponding function is *F_{SB4}: Continuous learning*.

Requirement AR_1 entails a function that provides tool selection recommendations (advice) to the user. It should assist the designer in smartly selecting the best matching and/or the most efficient data analytics tools for processing MoLD. This is an auxiliary function (F_{SA}), since it does not directly address a data transformation task (which is the purpose of basic functions). This smart auxiliary function is referred to as *F_{SA1}: Recommendation to support choosing task-relevant data analytics tools*. Requirement AR_2 can be fulfilled by a function that enables adaptation of the SDATB as a whole or of its specific operations based on individual users. This kind of adaptation can display specific operational options of the SDATB – for example, displaying only the user's proprietary databases or preferred tools. This smart auxiliary function was named *F_{SA2}: Adaptation to user*. Requirement AR_5 indicates that the SDATB should be connected everywhere and should provide the option of ubiquitous remote access. The corresponding auxiliary function was named *F_{SA5}: Permanent access*.

Requirement IR_1 formulates the need for identifying the user of the SDATB to offer him or her a secure and work-related environment. The corresponding smart interaction function (F_{SI}) has been named *F_{SI1}: Smart user identification*. This function has a basic function flavor in a different interpretation, since security and privacy are the most important requirements for a smart data management system. Requirement IR_2 prescribes a function that assists the designer in step-by-step navigation within the toolbox. Requirement IR_3 projects the need for a function that is always alert in monitoring designers' activities while the toolbox is in use. For this reason, requirements IR_2 and IR_3 can be combined in one smart interface function called *F_{SI2}: Monitoring help* that detects the status of the system and the workflow at every step and also monitors the activity history of the designers.

In making decisions on the possible scope of detailing and implementation of representative functions for the SDATB, we considered the assumable professional novelty and the practical significance. In addition, we had to consider the duration of the research project and the available capacities. Thus, we decided to include in the elaboration of the proposed demonstrative toolbox only those representative functions that most frequently occurred in the QBI and could be seen as core functions of a demonstrative data analytics toolbox. The chosen functions were F_{SB1}, F_{SA1}, and F_{SI1}. In the upcoming sections, all functions are detailed concerning their importance.

4.3. Underpinning the functions of the demonstrative data analytics toolbox

4.3.1. Collecting knowledge for the recommendation function for merging middle-of-life data streams

Data merging is one aspect of data management [104]. Traditionally, it is interpreted as the total of theories, techniques, and tools used to combine sensor data into a common representational format [105]. Its main purpose is to combine data from heterogeneous data sources [106]. It is widely used in many application domains, such as robotics, industrial manufacturing systems, smart buildings, and healthcare [107]. Data merging is a wide-ranging subject that gave root to many different terminologies, which are often used interchangeably [107]. It can be found in the literature under different names, such as data fusion, data consolidation, or entity resolution [108].

The ultimate objective of data merging is to improve the performance of a system by merging complementary and/or redundant information to reduce the uncertainties of measurements and to obtain information that cannot be perceived within one data source [109]. In some publications, the term information fusion is considered a synonym for data merging or fusion [110], whereas in other sources a clear distinction is made between them. Data merging is employed for raw data (not processed), and information fusion is used to define processed data [111]. Accordingly, information implies a higher semantic level than data [112]. Several types of data merging and fusion have been the focus of many research projects, such as decision fusion, data combination, data aggregation, sensor fusion, and multi-sensor data fusion [113].

In our project, we are interested in benefiting from processing MoLD. Consequently, the merging concerns data collected from product sensors while the product is in use. This type of merging is called multi-sensor data fusion [114] or multi-sensor data merging (MSDM) [115]. MSDM is rapidly evolving [116]. Its essence is combining data from multiple sources, and it helps provide access to information that cannot be provided by a single sensor or whose quality exceeds that of the information drawn from a single sensor [117]. Such technologies have replaced traditional information fusion systems, which involved user-owned and controlled sensor networks. In addition, they established systems and information architectures that are used for sensor tasking, data acquisition, fusion, dissemination, and decision-making [118]. It has been proved that using MSDM approaches is the only way to get the required amount of information with an expected level of intelligence [107]. MSDM approaches allow (i) an increased probability of detection, (ii) extension of spatial and temporal coverage, (iii) reduction of ambiguity, and (iv) improvement of system reliability and robustness [119].

Despite its multiple advantages, MSDM presents several challenges:

- Data imperfection, since sensor data are affected by a level of impreciseness and uncertainties in measurements [120].
- Outliers and counterfeit data, caused by the uncertainties in sensors (which originate in the impreciseness and the measurements noise) and by the ambiguities and inconsistencies in the environment [121].

- Conflicting data, because fusion of sensor data can be problematic when the fusion system is based on evidential belief reasoning [122].
- Data modality, created by the possible homogeneous and heterogeneous nature of sensor data [123].
- Data correlation, which is common in wireless sensor networks and can result in over-confidence or under-confidence in a data fusion algorithm if data dependencies are not accounted for [124].
- Data alignment and registration, which means that data collected from different sensors need to be transformed into a common frame before being fused, which may be influenced by calibration errors of individual sensor nodes [125].
- Data association, which may be both measurement-to-track and track-to-track association. The first one refers to the problem of identifying from which source a measurement originated. The second one deals with distinguishing and combining tracks and estimates the state of the same real-world target [126].
- Processing framework, which can be operationalized in either a centralized manner or a decentralized manner. The first one presents a communication burden, as all measurements must be transferred to a central processing node fusion. The second one does not suffer from this burden, as it allows each sensor node to locally process collected data [127].
- Operational timing – a crucial challenge given that the area covered by sensors may be large – in which the influential factors may vary in different rates. If this issue is not handled properly (especially in real-time applications), it can potentially degrade performance [128].
- Static versus dynamic phenomena, which means the observed phenomenon may be time variant or time invariant. In this case, the fusion process needs to incorporate measurement history [129], and it needs to determine how quickly the sensors capture the changes and update accordingly [130]. In this respect, it is important to check the validity of the fusion results.
- Data dimensionality, which concerns preprocessing data either locally (at each sensor node) or globally (at the fusion platform). Data are compressed into a lower dimension, which may result in a compression loss [131].

These challenges indicate that MSDM is a complicated task. In addition, MSDM can occur on three levels: low, middle, or high [132]. These levels refer successively to signal, feature, and decision levels [133]. Low-level merging applies to raw data coming directly from sensors and is used for knowledge construction or for cooperating with other nodes on complementary activities. It can be realized by a low-level abstraction or by performing local operations in a temporal domain [134]. Middle-level merging works with the features of datasets and flows and is thus often called feature-level fusion [135]. It is performed on preprocessed data or on information obtained by low-level merging of data. It can be realized by implementing feature extraction, pattern matching, or redundant computation operations [136]. High-level merging is a sophisticated process that is implemented by (i) performing semantic inference, (ii) executing complex reasoning, (iii) learning from and making decisions on sensor data, or (iv) exploiting cooperative patterns [137]. High-level data merging is computationally challenging and is difficult to realize for two reasons [62]. First, inferring semantic knowledge requires transforming a low-level representation of data and information into a higher-level representation. This transformation, however, typically suffers from the so-called information deficit. Second, having a system understand semantics assumes the system has (i) some manifestation of consciousness, (ii) a purpose, and (iii) an awareness of its surrounding and the state of knowledge. Needless to say, these characteristics are strongly related to human beings.

In our research, the objective was to elaborate a support function (F_{SB1}) for merging MoLD-Ss. It should merge computationally, semantically interrelated MoLD-Ss obtained from different sensors. In doing so, it helps the user gain additional information and knowledge from the streams to support decision-making in various contexts of product enhancement. After merging, the synthesized MoLD-S can be used to infer additional semantic information that initially was not conveyed by any one of the separate MoLD-Ss. Streams may complement each other and may provide additional semantic information when appropriate inference techniques are applied. The proposed function, F_{SB1}, adopts the principles of third-level MSDM. It contextualizes the information conveyed by MoLD-Ss and analyzes the information's meaning in that context. In fact, it extends to one level higher, since it generates recommendations about possible enhancements. This knowledge can be deduced by analyzing the merged data streams in a specific context and can be delivered to the designer as a displayed message. Such a function is a genuine enabler of the smartness of the SDATB.

4.3.2. Collecting knowledge for the recommendation function for choosing task-relevant data analytics tools

A recommendation system is a system that generates and delivers personalized, context-dependent suggestions to users by means of exploring a large space of alternatives or items [138]. There are two main types of recommendation generation: (i) content-based filtering (Figure 4.2) and (ii) collaboration-based filtering (Figure 4.3). The first type considers the previous preferences of the user and learns a preference model using feature-based representation of the content of recommendable items. The second type, collaborative filtering, is a technique that filters out items a user might like based on the reactions of similar users [139]. It relies on the identification of preference patterns in a community of similar users. It searches through a large group of people and finds a subset of users with tastes similar to that of a particular user. The filtering mechanism then identifies items that subset of users like and combines them to create a ranked list of suggestions. A so-called hybrid recommendation is a combination of the content-based filtering approach and the collaborative filtering approach (Figure 4.4). Its objective is to overcome the shortcomings of the constituents [140].

The specific objective of the function we are interested in is to recommend the most appropriate DATs for data processing for a designer in a given task context. The basis of recommendation is the information about the task the designer wants to accomplish and the list of the tools available to or known by the SDATB. Finding the recommendable tool starts with the specification of the task by the user and continues with the matching process conducted by the SDATB. In this way, the search time and efforts associated with tool selection are reduced. It is assumed that the recommendation function (i) performs semantic interpretation of designer's input, (ii) proposes or accepts a description of the task identified by the designer, (iii) makes inferences about the appropriateness of the available or known DATs, and (iv) recommends the best tools for the task at hand.

Clearly, various inputs and outputs must be defined to realize this function, as agreed upon principles and steps for generating a recommendation. As one input, the function requires information about the possible tasks of the designers (DT_x). These tasks DT_x are identified and stored in a structured manner in a warehouse of the SDATB. Additional descriptive information about the tasks is also stored. Similarly, the inventory of available or known tools is also stored in the SDATB warehouse. The correspondence between a chosen task (DT) and the possible tools, DAT_x, is analyzed in the process of matching. The tools are to be ranked according to their matching and the one most appropriate for the

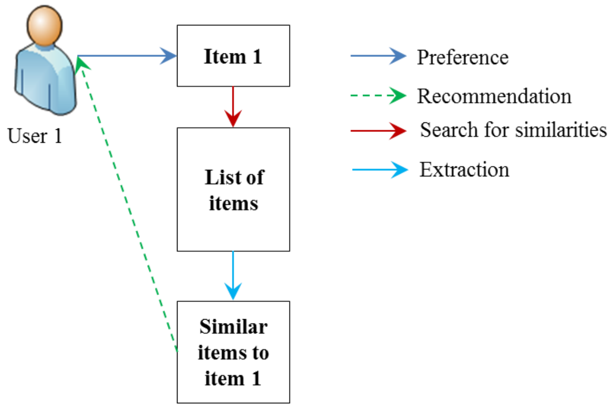


Figure 4.2. Content-based filtering recommendation

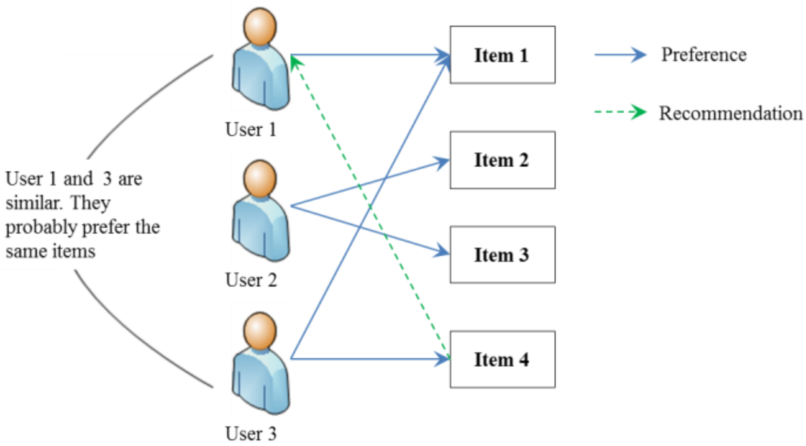


Figure 4.3. Collaborative filtering recommendation

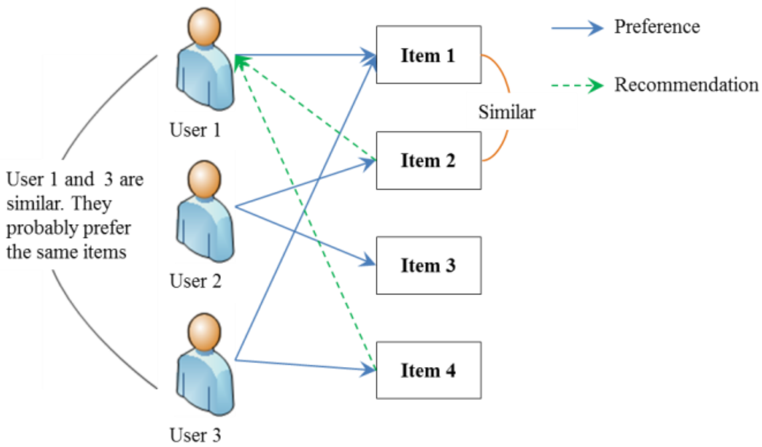
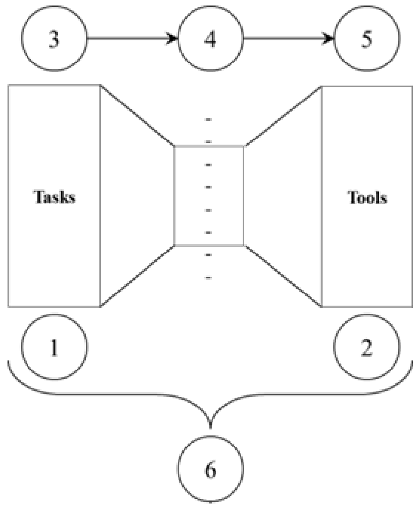


Figure 4.4. Hybrid recommendation

designer's purpose is to be displayed to designer.

The related questions from a computational point of view are these: (i) If the two sets (DT_x and DAT_x,) are provided, how are the matching ones found? (ii) If there are multiple possibilities, how is the best DAT match for a given DT selected? Dual-criteria matching has been used as one methodological approach. Figure 4.5 shows the concept of the realization of the recommendation function. We have assumed that matching can be based on the identification of similar features of the tasks and the tools. Three of them have been specified: (i) the source of the dataset to be analyzed (DS), (ii) the category of the analyzed dataset (DC), and (iii) output of data analytics (DO). The evaluation of these criteria provides an indicator of the matching between the feature matrixes of DAT_x and DT_x. The criterion for being selected is the cardinality of the shared features.



(1) DT_x,
(2) DAT_x warehouse,
(3) DT_x identification,
(4) Characterization of DT_x and DAT_x,
(5) Matching DT_x and DAT_x,
(6) Selection of probable DAT_x is offered

Figure 4.5. Principle of realizing the recommendation function

For the task of ranking or choosing the best tool from DAT_x, the expected performance of each tool provides information. Three criteria have been identified for how the tool works: (i) graphical capabilities (C_1), (ii) speed of computation (C_2), and (iii) computational performance (C_3). The fulfilment of these criteria should be weighted. To this end, a matrix of weights (W) has been allocated to all DAT_x in terms of C_x. The (transpose) vector of weights is represented as W*(WC_1, WC_2, WC_3). This helps the SDATB rank the available and known tools based on best and worst performances and propose the one with the best computational performance to the designer. This reasoning logic and these computational elements have been used in the computational realization of the tool recommendation function, as discussed in detail in Section 4.4.2 below.

4.3.3. Collecting knowledge for the smart user identification function

Many identification approaches, also known as authentication mechanisms, have been proposed for identifying users in various systems. User identification is the process used to verify the authenticity of a user in terms of his or her access to information [141]. It is mainly used for protecting information from illegal access [142], but it can also be used to set up a customized work environment for the user. In some cases, it is capable of improving the flow of information and tracking system [143]. Systems capable of user authentication can use standard devices such as keyboards and mice or special devices such as web cameras or tracking devices. Our concern is novel smart techniques for identification. Accordingly, we did an explorative study with this is mind. It must be mentioned that we use the word "identification" to refer to the fact of being identified and the word “authentication” to refer to the act or process of identifying someone (or something).

User identification can be done via (i) password authentication [144], (ii) biometric authentication [145], (iii) two-factor authentication [146], (iv) identity-based authentication [147], or (v) role-based authentication [148]. Password authentication is one of the simplest and most convenient authentication mechanisms [149]. It can be divided into two types; one requires a weak (easy to guess) password and the other requires a strong password (one that is complicated in terms of its components) [150]. Biometric authentication allows both verification and actual identification. It can be divided into two classes: (i) physical and (ii) behavioral [151]. Physical authentication is linked to body shape, which differs from person to person. It can be done via fingerprints, face recognition, hand geometry, or iris recognition. Behavioral authentication is related to the person’s behavior: for example, signature, keystroke dynamics, or voice (which can also be seen as physiological).

Two-factor authentication is defined as an authentication mechanism that uses more than one factor (for example, a combination of a password and a chip card) in authenticating the user [152]. It is widely used. Identity-based authentication allows identification by name, public key, or serial number [153]. It is sometimes confused with role-based authentication, but role-based authentication uses role credentials, rather than the identity indicators of the user, for authentication without disclosing the user’s identity information. The system verifies the user’s role credential and checks whether the relationship between user and role is correct [154].

The objective of the smart user identification function (F_{SI1}) is to control system user access to the system resources and to offer a secure environment in which the designer can analyze data. This function can also support setting up the whole working environment according to the preferences of a particular user. Consequently, the smart identification includes both the verification of the identity and authentication of the designers. Since two-factor authentication is more secure than authentication using only one factor (or method) [155], we decided to build two-factor authentication using biometric identification. The biometric identification is done using face detection and face recognition. To make the function secure, the computer’s integrated or external camera needs to detect that only one face is present. If two or more faces are detected, the identification will not happen.

To augment the security of the SDATB, the proposed function F_{SI1} allows recording the identification history of a particular designer, including the date and time of past identifications. Based on this, the SDATB and/or the user can check whether hacking might have occurred. In such situations, the user will be able to reset the identification indicator, the history of attendance, and the password, as it now happens routinely. For the identification, the designer’s laptop or desktop must have an internal or external camera. For computational implementation of the function F_{SI1}, the principles and steps of the identification must be chosen, and the necessary inputs and provided outputs must be defined. The assumed input is the image information the camera provides of the designer’s face. The SDATB captures the face image information and matches it to existing saved images. If the designer’s face is recognized, the entry process commences. The designer will receive a welcome message and permission to access all

SDATB resources and databases. Users can also recall their attendance history. New users will be asked for personal details, and a user profile will be created before they are given access.

4.4. Elaboration of the representative functions of the demonstrative data analytics toolbox

The computational functions of the SDATB are specified below. This process is often called functional decomposition. The objective of the decomposition is to identify the subordinate functions that are needed to achieve a particular function. In this way, the main functions are decomposed into multiple levels of sub-functions, and the lowest-level sub-functions are further decomposed into elementary functions [156]. Functional decomposition supports the development of the computational algorithms and codes.

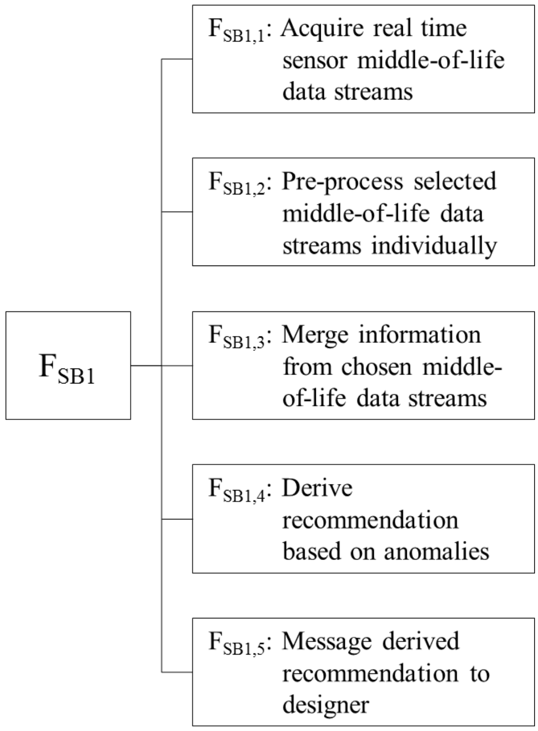


Figure 4.6. High-level functional decomposition of F_{SB1}

4.4.1. Elaboration of the recommendation function for merging middle-of-life data streams

The computational implementation of the function F_{SB1} for merging data streams has three elements. Symbolically, they can be expressed as

$$F_{SB1} = F_{SB1}\,(DI_{SB1}, CP_{SB1}, SO_{SB1}), \tag{4.1}$$

where F_{SB1} is the basic function providing recommendations for merging MoLD-Ss, DI_{SB1} are the inputs by the designer (MoLD-Ss), CP_{SB1} is the computational mechanism, and SO_{SB1} represents the outputs expected from the SDATB after execution of F_{SB1} (i.e. messages displayed to the designer about the results of the data stream merging). The necessary computational procedures can be defined if F_{SB1}, as a main basic function, is decomposed to lower-level functions and related requirements are considered. The intermediate lower-level functions of F_{SB1} were already presented in Section 4.3.1. The underlying process is as follows. First, the SDATB acquires the MoLD-Ss selected for merging from the corresponding sensors in real time. Then, after the designer choses streams for further analysis, those streams are preprocessed individually based on their data. The preprocessed MoLD-Ss are then fused together. The following step focuses on detecting anomalies in the merged data streams and determines what might be wrong with the product based on data. Once the meaning is given to the fused MoLD-Ss, the SDATB derives recommendations on what should be done with the product (such as enhancement possibilities). Finally, this recommendation is sent to the designer as a message appearing on the screen.

The procedural steps of merging MoLD-Ss (function F_{SB1}) are shown in Figure 4.6. This function decomposes to five sub-functions. A lower-level functional decomposition of F_{SB1} is summarized in Figure 4.7. We have assumed that the designer specifies for the SDATB what data streams will need to be merged. Another assumption is that only temporally finite data streams are handled by the SDATB. This latter assumption facilitates the application of machine learning. The sub-function $F_{SB1,1}$ locates the considered sensors on the product and forwards the data streams provided by them to the SDATB. Our assumption is that the forwarded MoLD-Ss may be stored not only on the background storage devices of the SDATB host computer but also on a separate storage device. To get a reconfirmation from the designer, sub-function $F_{SB1,2}$ presents the data streams to the designer using various means to visualize the MoLD-Ss (for example, plots or histograms). In addition, the sub-function preprocesses the single-modality data streams by selecting particular processing rules. As an example, some rules can eliminate parts or the whole of redundant data streams that are not likely to affect the merging. To avoid the need to transfer and process vast amounts of idle information, the rules may operationalize up/down sampling of values, value transformation, and reducing noise in the data to decrease unnecessary variance of the data to be processed. The sub-function $F_{SB1,2}$ applies a kind of configured data processing, which is required because of the time-consuming nature of processing the data. In this context, “configured” indicates that, for complicated data streams with unknown patterns, comprehensive structural preprocessing (filtering or ordering) is applied, whereas for less complicated data streams, preprocessing is simply data normalization.

The computational merging of data streams is done by sub-function $F_{SB1,3}$. The principle of fusion is correlation based on the time stamps of data in the streams. First, the sub-function generates intermediate representations to reduce time-dependent data to a compact fixed-length vector. Then it combines the data streams and generates a behavior descriptor based on the merged MoLD-Ss. To facilitate the application of machine learning, sub-function $F_{SB1,3}$ embeds the fused sensor data streams into a so-called latent space (also called a hidden space). In this space, data is mapped in such a manner that similar data points are close to each other. In the case of neural network-based machine learning, features are extracted through a number of layers of the network architecture, and the operation (function) that maps the input before the last layer projects into the latent space. In other words, the features lie in the latent space. The latent space representations can be used to transform complex forms of raw data into simpler forms that are easier to analyze. Mapping to the latent space also helps in clustering similar cases.

The data streams may contain anomalies regarding the operation of the product in question. The sub-function $F_{SB1,4}$ (i) detects anomalies in the merged data streams, (ii) matches the anomalies to pre-programed knowledge in the SDATB, (iii) orders the anomalies based on their similarity, (iv) makes a report on all of the ordered anomalies based on the merged MoLD-Ss, and (v) converts the outcome into a specific recommendation. The last sub-function, $F_{SB1,5}$, (i) retrieves a template for message construction, (ii) constructs a message for the designer according to the recommendation, (iii) uses the retrieved template to construct the message to be delivered to the designer, and (iv) communicates the message to the designer relating

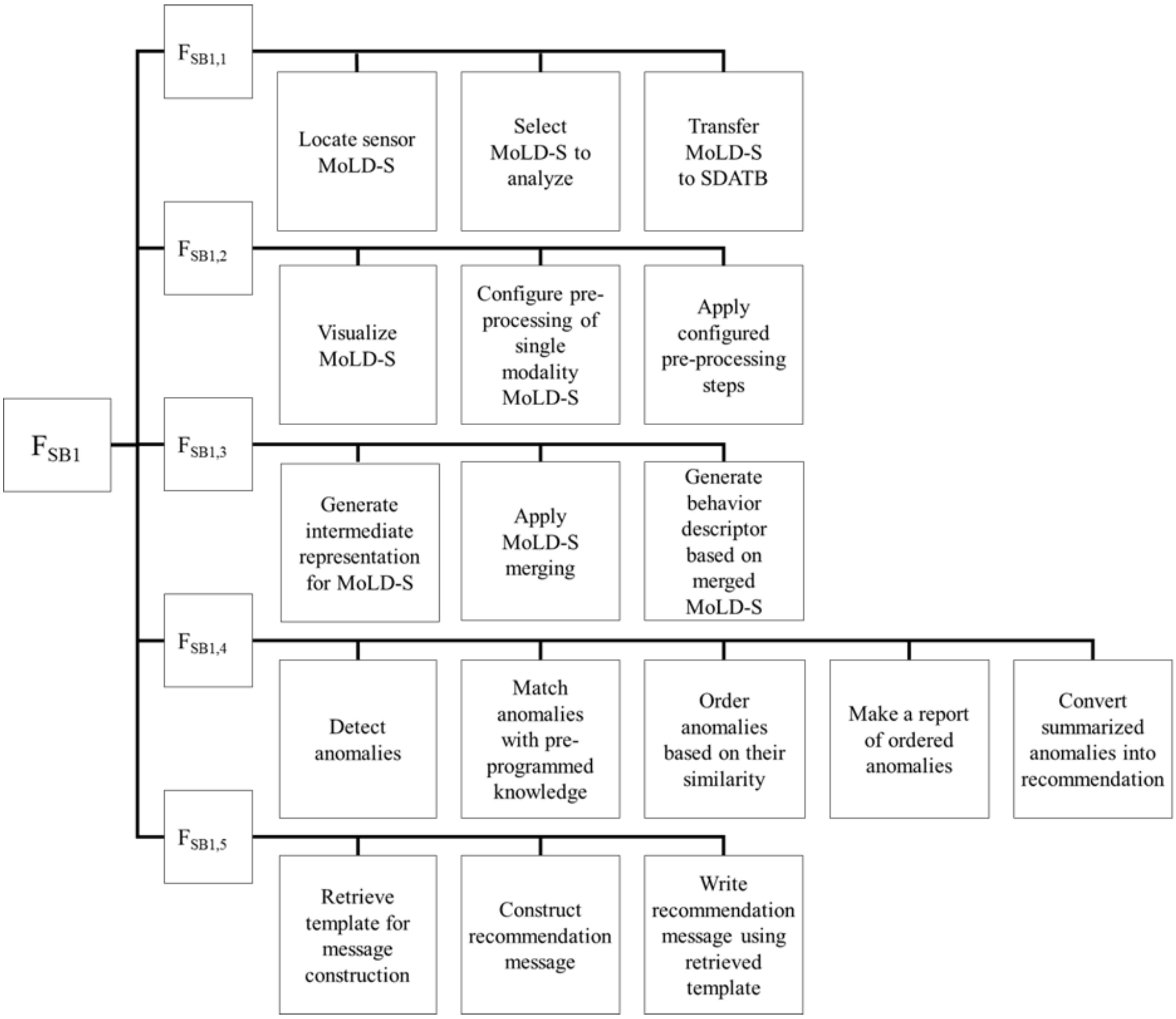


Figure 4.7. Low-level functional decomposition of F_{SB1}

what is improper with the product according to the merged data. In the case of the SDATB, this can also be followed by a recommendation for actions to take to solve detected anomalies in product operation, although this step is not indicated in Figure 4.7.

To realize the function of $F_{SB1,1}$, the SDATB needs to (i) locate the sensors producing the MoLD-Ss, (ii) identify and access the data streams to be merged, and (iii) import these streams from their storage place (for example, the cloud) to the SDATB. Moving MoLD-Ss from external storage into the SDATB is a common procedure (several commercial tools allow retrieval of data streams from external storage). However, current software tools do not allow the collection of real-time data streams. Consequently, realization of $F_{SB1,1}$ requires the development of new algorithms. Since we had no opportunity to have access to an appropriate sensor network and the multiple data streams generated by its nodes, we provided the necessary data files using computational simulation. The constructed files were used both in the algorithm development stage and in the validation of the algorithms. What it means is that we are not dealing with function $F_{SB1,1}$ here, since the data streams have been included in the toolbox database directly. We assumed that, in a real-life situation, the SDATB would have access to sensors and would be able to receive multiple MoLD-Ss. These data streams are checked before they are used for merging. The SDATB offers the option of visualizing all data streams received, and the designer can select varying numbers of the MoLD-Ss for analysis and merging by the SDATB. The basis of merging is the "internal" affordance (semantic cohesion potential). Anomalies detected in the fused streams are identified and included in the results.

4.4.2. Elaboration of the recommendation function for choosing task-relevant data analytics tools

The computational implementation of function F_{SA1} for choosing task-relevant data analytics tools can be expressed symbolically as

$$F_{SA1} = F_{SA1}\,(DI_{SA1}, CP_{SA1}, SO_{SA1}), \tag{4.2}$$

where F_{SA1} is the auxiliary function for providing recommendations for merging MoLD-Ss, DI_{SA1} are the inputs by the designer (the task specification), CP_{SA1} is the computational mechanism, and SO_{SA1} are the outputs expected from the SDATB after execution of F_{SA1}. The output is a finite number of possible DATs, or the best match, recommended to the designer. To be able to make a recommendation, the SDATB should recall several related information constructs stored in its database. These are referred to as system inputs (I_x) and include the following: (i) I_1 = set of DTs, (ii) I_2 = set of DSTs, (iii) I_3 = library of DATs, (iv) I_4 = descriptions of DATs, (v) I_5 = descriptions of DSTs, and (vi) I_6 = matrix of weights of DATs.

The set of DSTs incorporates the possible subtasks of the designers. In principle, the tasks of the designer may be abundant, and some of the tasks might be not yet supported by the SDATB in its early stages. Furthermore, the specification of the actual task by the designer might be incomplete, insufficient, or not sufficiently concrete. For these reasons, the set of DSTs includes all subtasks that can be managed by the SDATB. In our research, we applied the concept of look-up tables to associate every element of the set of DSTs with one or more appropriate DTs. Compilation of the set of DSTs increases the chances of proper interpretation of the designer's tasks and of providing dependable recommendations.

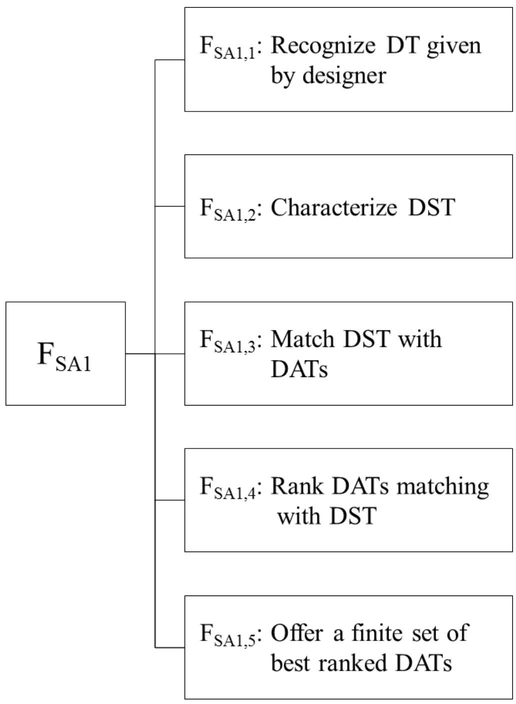


Figure 4.8. High-level functional decomposition of F_{SA1}

The sub-functions needed for the realization of function F_{SA1} are shown in Figure 4.8. Sub-function $F_{SA1,1}$ recognizes the design task given by the designer. Based on the input obtained from the designers, it determines the task features that allow a preliminary mapping of an incomplete task specification to the stored formal and complete data analysis task specifications. If the mapping is successful, then the stored formal task specification is returned to the user for approval and is used in the follow up computational actions. Sub-function $F_{SA1,2}$ characterizes the confirmed DST by designer and creates a vector of subtask including its characteristics. The mapping of the data analytics tools and tasks is done by sub-function $F_{SA1,3.}$ It selects the candidate DATs based on the task features. The tool features and the task features are represented as two vectors that can be browsed and compared. Sub-function $F_{SA1,4}$ calculates a syntactic distance between the feature vectors of the DST and the selected DTs that it uses to rank them. The candidate DTs with the shortest syntactic distances will be put on top of the list of candidates. Sub-function $F_{SA1,5}$ shortlists the candidate tools and presents them to the designers, together with an explanation of each tool. Figure 4.9 shows one lower-level decomposition of the sub-functions $F_{SA1,x}$.

After recognizing the DT given by the designer and building its corresponding vector, the system needs to build similar vectors for all DATs. Their contents can be used to calculate the distances between the respective features of the DST vector and the DAT vectors. The tools presenting minimum distances between the features will be retrieved. To facilitate the ranking of the tools after selection, the weights (Ws) allocated to features of the selected tools are also considered. These pieces of information related to the tools are used as inputs for the fourth sub-function ($F_{SA1,4}$), which ranks DATs based on their weights. The sum of weights for C_1, C_2, and C_3 of every tool must be calculated and then sorted from high to low value. This is done by calculating $WC_1 + WC_2 + WC_3$ for each tool and ordering the obtained values in descending order. In this way, it is

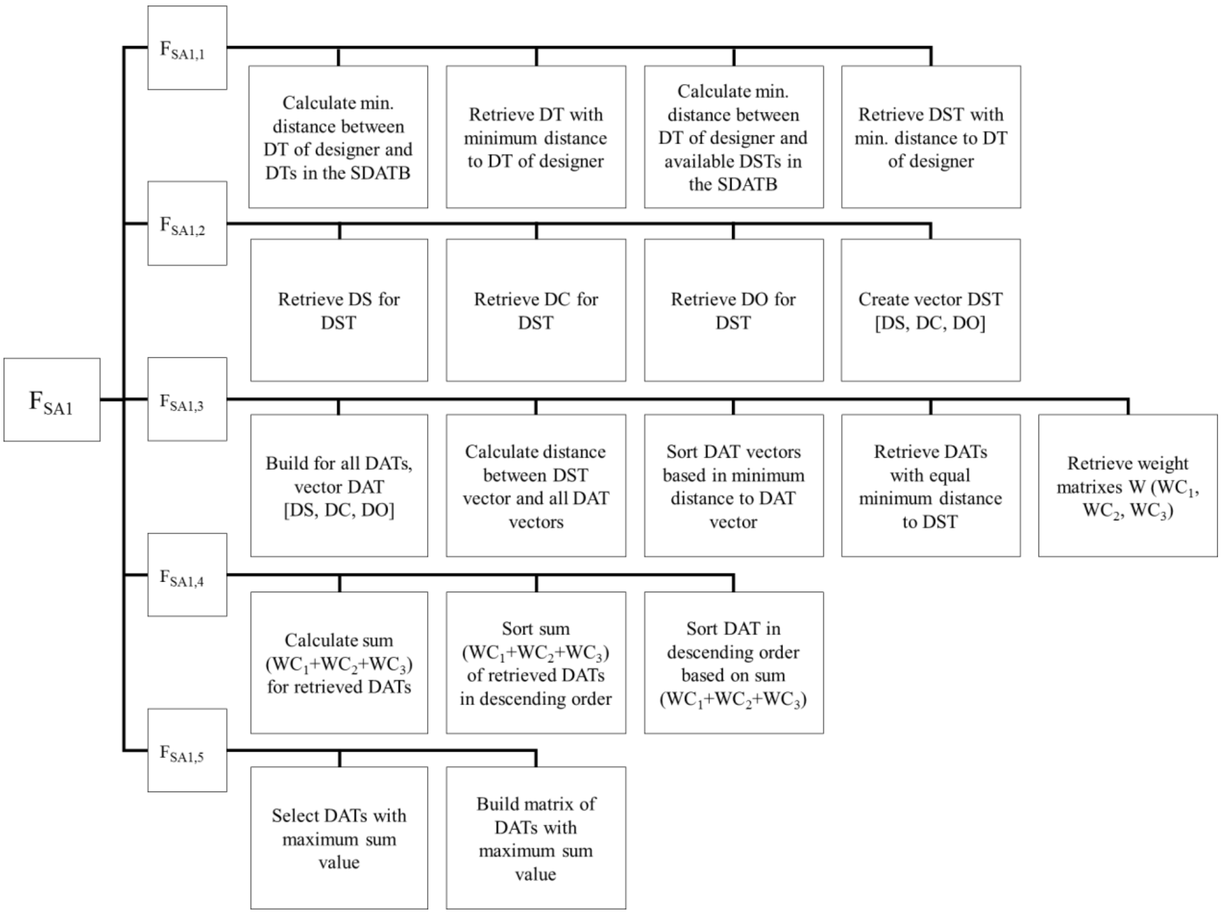


Figure 4.9. Low-level functional decomposition of F_{SA1}

possible to rank the tools from the highest-degree match with DTs to the lowest-degree match. The resultant ranking of DATs is used as input for the fifth sub-function ($F_{SA1,5}$), which offers a finite list of ranked tools. To achieve this output, the tools with maximum sum values are selected, and a final matrix of the best-matching DATs is generated.

4.4.3. Elaboration of the function for smart user identification

The computational implementation of the main function F_{SI1} for smart user identification can be expressed symbolically as follows:

$$F_{SI1} = F_{SI1}(DI_{SI1}, DI_{SI2}, CP_{SI1}, SO_{SI1}, SO_{SI2}, SO_{SI3}), \tag{4.3}$$

where F_{SI1} is the interface function for smart user identification, DI_{SI1} is input by the designer (name of designer), DI_{SI2} is input by the designer (face of designer to be detected by the camera), CP_{SI1} is the computational mechanism, SO_{SAI1} is output by the SDATB (secure access), SO_{SAI2} is output by the SDATB (history of designer's attendance), and SO_{SAI3} is output by the SDATB (designer's personal profile in the database).

To realize the main interface function F_{SI1}, the intermediate-level and lowest-level functions should be determined. Details of the intermediate-level functions were introduced in Section 4.3.3. We must note that, although face recognition is a standard solution in safety-critical and image processing applications, using it in the context of the SDATB is rather new. The sub-functions were specified according to the regular process of face recognition. This includes the following activities: (i) setting up the camera for imaging, (ii) detecting the face, (iii) analyzing and recognizing the face image, (iv) storing the face image and recording the attendance of the designer in the database, and (v) providing access to SDATB services. The high-level functional decomposition of function F_{SI1} is shown in Figure 4.10. Figure 4.11 shows the lower-level functional decomposition of F_{SI1}.

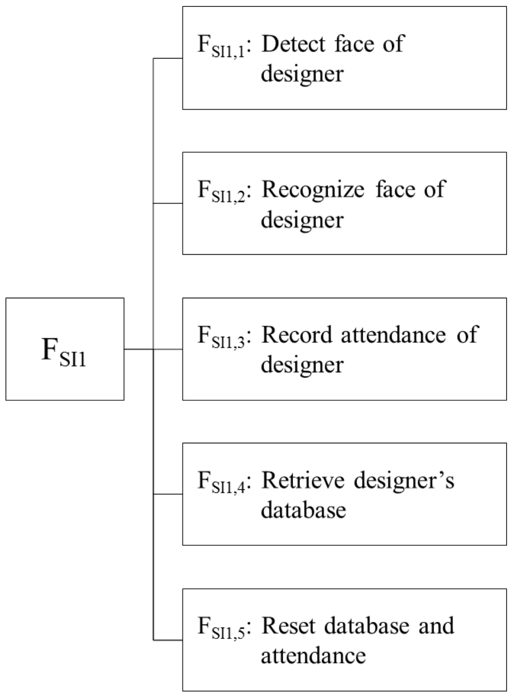


Figure 4.10. High-level functional decomposition of F_{SI1}

The sub-function $F_{SI1,1}$ of the SDATB locates the image source (camera) on the designer's laptop and acquires the captured image. To facilitate the extraction of the face from the image, the colors are converted to a grayscale representation. Once the face is extracted, it is saved in the SDATB database. This face image will be used as input for $F_{SI1,2}$, which applies machine learning to recognize the face. A machine learning algorithm must be trained for this purpose and to match the face image with other face images stored in the database.

The recognized face is annotated with the name of the designer and saved in the designer's user profile, which resides in the database. When the user is identified for the

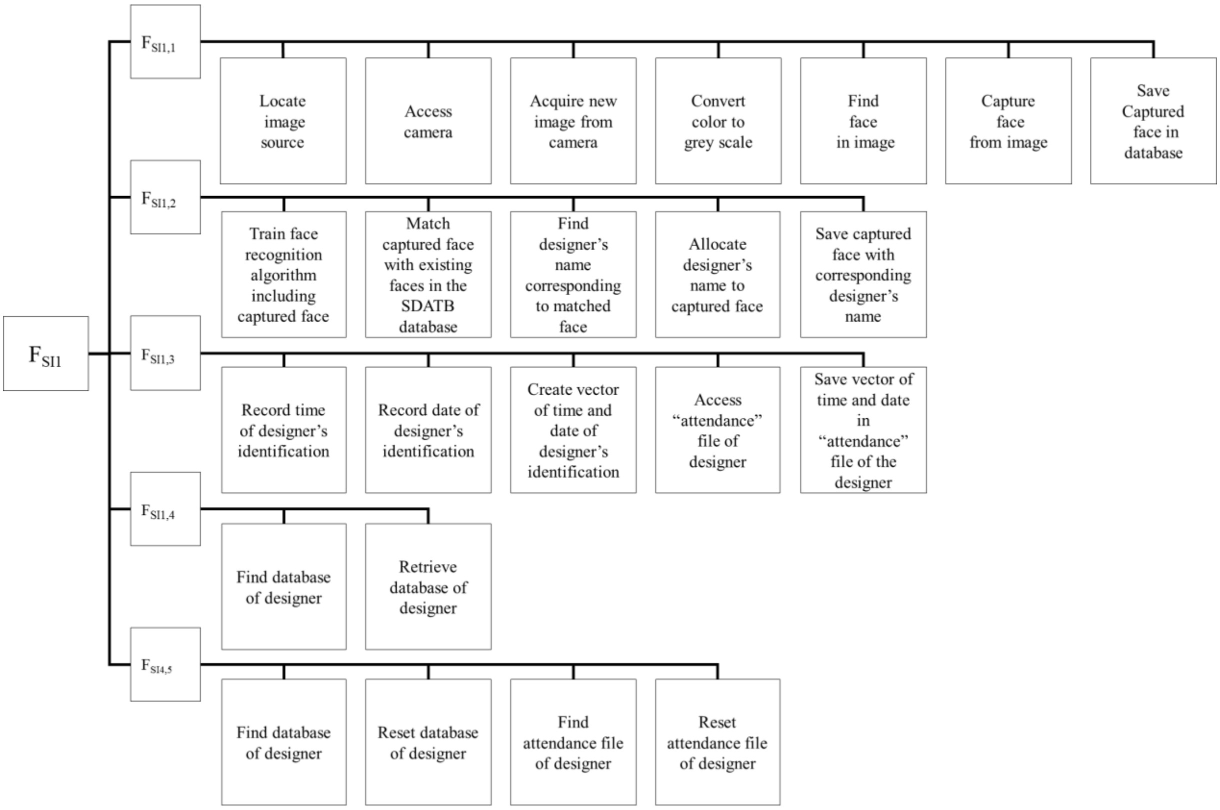


Figure 4.11. Low-level functional decomposition of F_{SI1}

first time, a log-in document called "attendance" is created as a digital file. The sub-function $F_{SI1,3}$, completes the user identification with the date and time of access and saves these pieces of information in the attendance file. Each time the user is identified and given access, that information is recorded in this file. After the completion of the authentication process, the designer has access to all system resources. Optionally, the system may set up the designer's last working state. The sub-function $F_{SI1,4}$ finds and retrieves the designer's personal identification file from the database. The sub-function $F_{SI1,5}$ gives the user the option of resetting the database and managing all past attendance records.

4.5. Architecting the modules of the demonstrative smart data analytics toolbox

4.5.1. Architecture of the merging of middle-of-life data streams recommendation module

Form a software engineering point of view, the main functions of the SDATB are provided by various modules. Specification of the modules and determining the computational algorithms included in the modules is the task of architecting. For instance, two external components are needed for the architecture of the main function F_{SB1}. One of them is a system user interface, which enables communication between the designer and the SDATB. It also transfers the inputs and outputs to and from the toolbox. Another component is the database, also referred to as the knowledge warehouse. In addition to data, it stores the rules and conditions for analyses, as well as the results of merged data streams.

Figure 4.12 shows the overall architecture of the MoLD-Ss merging module of the SDATB. The main constituents are (i) the search engine, (ii) the database, (iii) the preprocessing unit, (iv) the merging unit, (v) the recommendation unit, (vi) the explorer, (vii) the query manager, and (viii) the user interface. The lower-level components of the units are shown in Figure 4.12. The MoLD-Ss explorer, used for exploring the data streams to be analyzed, is a kind of entry point to this module. The MoLD-Ss preprocessing unit communicates with the designer and receives and processes the individual MoLD-Ss in the toolbox. The MoLD-Ss manager visualizes the data streams stored in the database and makes them available for the search engine. The preprocessing configurator determines the preprocessing rules and conditions to be applied to the individual streams by the preprocessing executor. These two components use knowledge already existing in the database.

The preprocessed MoLD-Ss are transferred to the merging unit, which is composed of four components: (i) the merging executor, (ii) the anomalies detector, (iii) the semantic similarity calculator, and (iv) the anomalies organizer. These components are closely related to the knowledge stored in the database. The semantic similarity calculator compares the explored anomalies with those stored to determine resemblances. The anomalies organizer manages the weights and filters and organizes the anomalies to be used by the recommendation unit. The recommender agent converts the information generated by the above components into recommendation contents. The message generator produces messages to the designer using the recommendation contents. Finally, the query manager converts the produced message to human language and communicates it to the designer as a recommendation via the user interface.

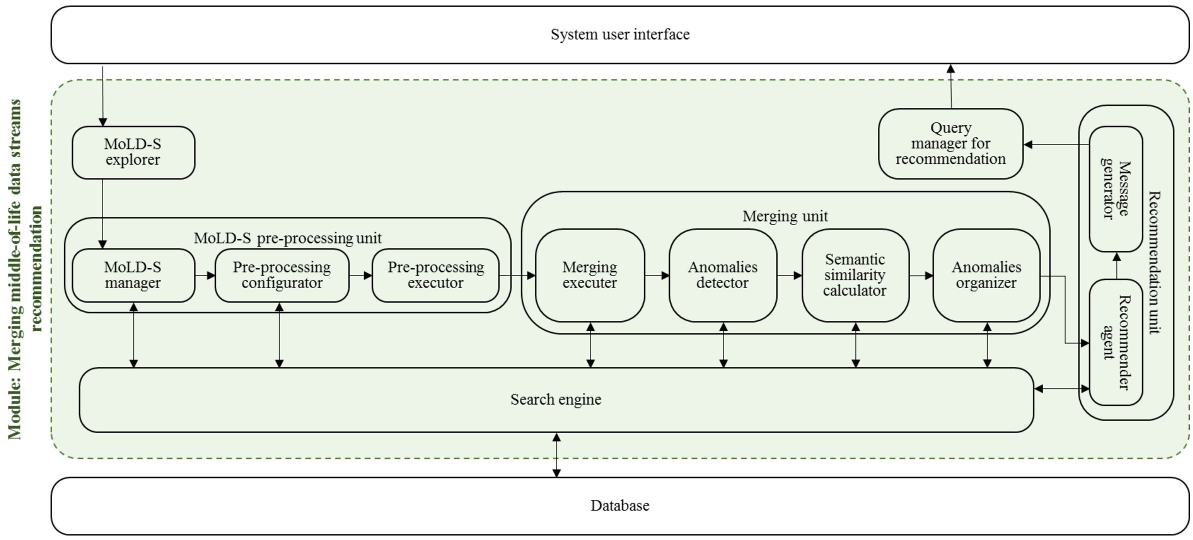


Figure 4.12. The overall architecture of the recommendation module for merging middle-of-life data streams

4.5.2. Architecture of the task-relevant data analytics tools recommendation module

The overall architecture of the module implementing the main function F_{SA1} is shown in Figure 4.13. The main constituents are as follows: (i) the search engine, (ii) the database, (iii) the query manager, (iv) the task manager, (v) the tools manager, (vi) the similarity calculator, (vii) the subtasks explorer, (viii) the recommendation agent, and (ix) the system user interface. Being of the same nature, the DTs and DSTs are handled by one manager. A separate manager handles the data analytics tools. The architecture includes the input "gate" as the system user interface of the SDATB: the DT is specified and a finite set of DATs is sent back to the designer.

The design task specified by the designer is transferred from the interface to the design task explorer for identification. The similarity calculator uses the search engine to compare it with DTs already stored in the database. In the case of a successful match, the query task manager sends the formal task specification to the interface for interpretation of the task. It follows the same procedure to identify the DST. The subtask descriptor characterizes the DSTs to be used in the tools selection. This characterization is used by the DATs explorer in matching. The explorer searches for tools in the database via the search engine and sends them to the tools descriptor to look for tools adequate for the DSTs. Based on the sent results, the recommender agent chooses the best DATs to propose. The query manager interprets this output on DATs and sends the information about the best matching data analytics tools to the designer through the user interface.

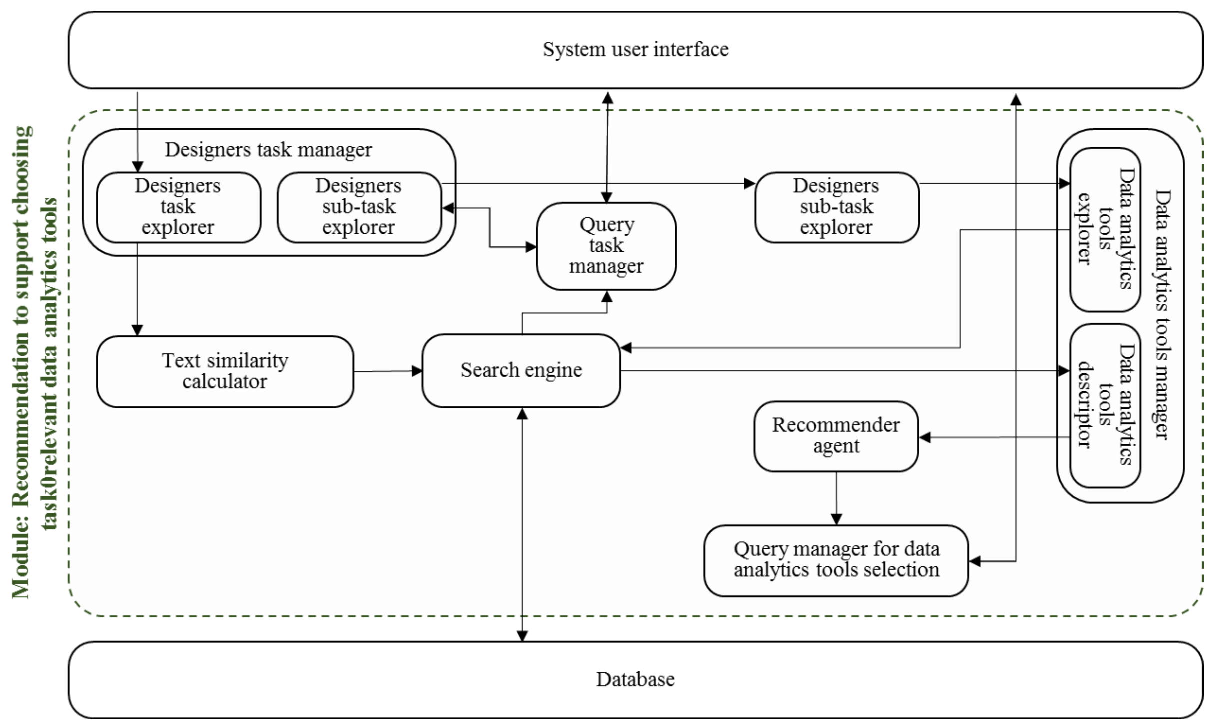


Figure 4.13. The overall architecture of the recommendation module for choosing task-relevant data analytics tools

4.5.3. Architecture of the smart user identification module

The overall architecture of the module performing the interface function F_{SI1} is shown in Figure 4.14. The constituents of the module are (i) the image acquisition unit, (ii) the face recognizer unit, (iii) the search engine, (iv) the user profile manager, and (v) the system user interface. The image acquisition unit consists of (i) the camera monitor, (ii) the image adjuster, and (iii) the face detector components. The face recognizer unit includes (i) the feature extractor, (ii) the face identifier, and (iii) the face recognition manager components. The database of the SDATB plays a key role in the architecture by storing (i) the user names, (ii) the training data, (iii) the attendance documents, and (iv) the user profiles. Different manager units are needed to assure interaction within the SDATB.

The camera monitor establishes access to and activates the computer's camera. The image adjuster aligns the image detected by the camera and converts its colors. The face detector extracts face image information from this converted image and saves the result in the database. The facial feature extractor selects and processes the facial features, which are sent to the face identifier. This component determines whether the captured face image matches any image stored in the SDATB database. The face identifier is in direct connection with the face recognition query manager, which generates messages to the designer. These messages are converted into human-readable format and communicated to the designer via the system user interface. The designer can have access to his or her personal profile when the two-factors identification is completed. The profiles query manager is connected to the database and generates the attendance document via the search engine connected to the SDATB database.

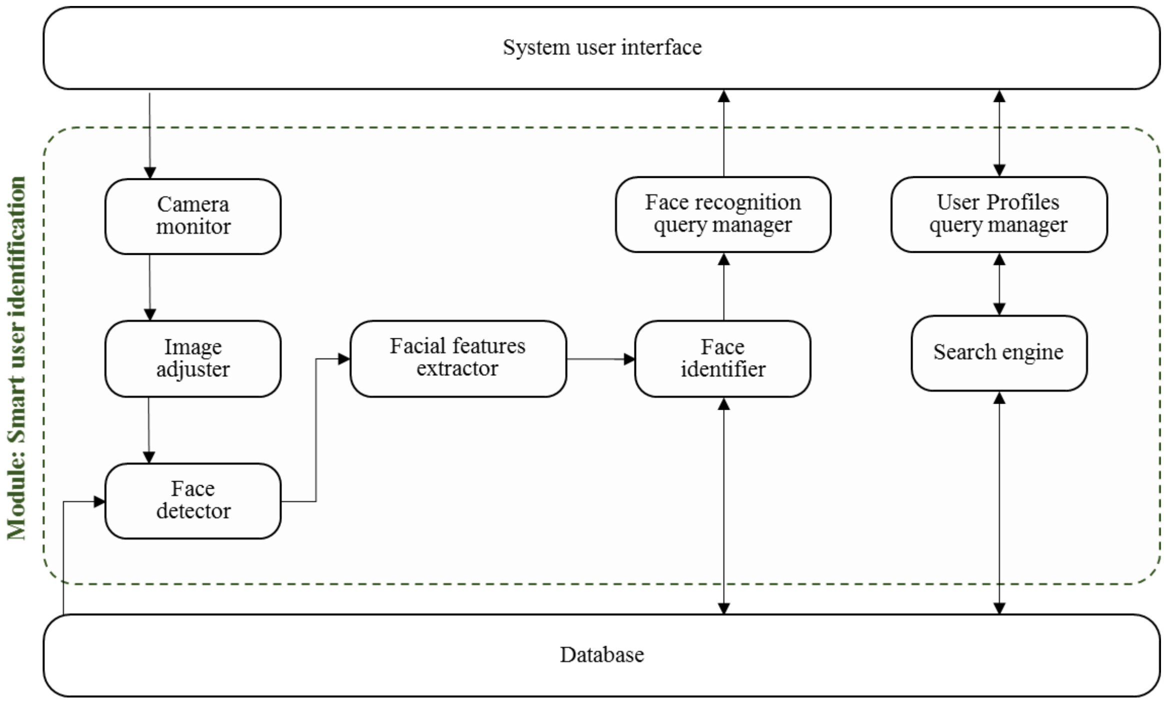


Figure 4.14. The overall architecture of the smart user identification module

4.6. Summary of the work and reflection on the results

In this chapter, the conceptualizations of three representative functions of the demonstrative SDATB were discussed. The emphasis in research was on the functional and architectural specification of the representative functions. The work comprised five larger steps.

As a first step, we conducted a literature study to aggregate information and facilitate the ideation process. We analyzed the findings together with the outcomes of the QBI and the ATF process, reasoned about the feasibility of each option, and examined the technical affordances related to them. Five knowledge domains associated with smart data analytics tools were studied: (i) interfacing, (ii) data management, (iii) reasoning, (iv) learning, and (v) analytics methods and techniques. Our major findings were that the research efforts reported in the literature were mainly related to (i) processing performance, (ii) approaches to big data handling, (iii) big data storage within data analytics tools, and (iv) the need for the manifestation of smartness within DATs.

As a second step, we transcribed the theory obtained by ATF into requirements for an SDATB. Four types of requirements were synthesized: (i) general requirements, (ii) BRs, (iii) ARs, and (iv) IRs. The general requirements guaranteed the logic and the procedural correctness within the SDATB. They concerned, for instance, the robustness of the computation, the understanding of the inputs given by the designer, the functional connectivity between the constituents of the toolbox, and the correctness of the interactions between the designer and the SDATB. The BRs were related to data processing within the toolbox. They implied that the SDATB should (i) fuse MoLD-Ss, (ii) recognize patterns in a context, (iii) build situation awareness, (iv) reason from past cases, and (v) learn from previous applications. The ARs concerned the management and transformation of data inside and outside the toolbox. They stipulated that the SDATB should (i) allow tools recommendation, (ii) be tailored to user needs, (iii) enable users to merge qualitative and quantitative MoLD, (iv) provide a dynamic and high-volume storage capacity, and (v) be permanently accessible. The IRs, as their name implies, were focused on the manifestation and the quality of the interface between the system and the user. They indicated that the SDATB should (i) authenticate and identify designers, (ii) help the designer navigate, (iii) allow process monitoring, and (iv) include a variety of dynamic visualization options.

The third step specified computational functions of the SDATB, starting from the basic, auxiliary, and interface requirements. We primarily considered the fulfilment of those novel requirements that had not yet been addressed and studied in the literature in the context of an SDATB. Consequently, three-three BRs, ARs, and IRs were transformed into concrete functions to be included in the SDATB. The functions realized for the BRs are (i) recommendations for merging MoLD-Ss, (ii) semantic interpretation of MoLD, and (iii) continuous learning. The computational functions implied by ARs are (i) recommendations to support designers in choosing task-relevant data analytics tools, (ii) adaptation of system smartness to the given user, and (iii) provision of permanent remote access to services. The functions addressing the IRs are (i) smart user identification and (ii) context-sensitive monitoring help (derived based on two IRs).

The fourth step included the ideation and detailing of the representative functions for the demonstrative SDATB. From the above set of requirements, we chose one-one representatives (1BR, 1AR, and 1IR) to be realized as computational functionality for the demonstrative SDATB. Being a main basis function, the first elaborated function was "recommendation for merging middle-of-life data streams." The second was the main auxiliary function "recommendation to support choosing task-relevant data analytics tools," and the third was the main interface function "smart user identification." We specified inputs and outputs related to all computational functions as well as their decomposition from a high level to an intermediate level. The first function is for merging originally interrelated MoLD-Ss and making recommendations to the designer based on the merged streams. By combining data streams, the designer may obtain additional information that is not conveyed by any one of the original data streams. This semantic addition may support product and service enhancement and may improve the designer's decision-making while reducing designer time and effort. The second function recommends to a designer the most appropriate DATs for data processing in a given task. It reduces designer time and effort in selecting tools and compensates for the designer's lack of knowledge concerning novel DATs. It offers a semantic interpretation of designer's input, proposes a description of the problem identified by the designer, reasons with a large number of DATs, and recommends matching tools for the designer's task. The third function offers convenient personal identification, user history management, and a secure environment for designers via a face recognition process that records designer attendance and stores each designer's identification in the database.

The final step developed architectural specifications for the modules that merge MoLD-Ss, choose task-relevant data analytics tools, and perform user identification. The needed constituents of the modules were clarified, as were their functional and structural connections. The architecture of the modules has been overlaid by the computational workflow they implement. The results generated in this phase of the research work serve as input for the next research cycle, which focuses on the elaboration and implementation of the needed algorithms and the related data constructs.

4.7. References

[1]. Triplet, T., & Butler, G. (2012). The EnzymeTracker: an open-source laboratory information management system for sample tracking. *BMC Bioinformatics*, 13(1), 15.

[2]. Saldivar, A.A.F., Goh, C., Li, Y., Chen, Y., & Yu, H. (2016). Identifying smart design attributes for Industry 4.0 customization using a clustering genetic algorithm. In: *Proceedings of the 22nd International Conference on Automation and Computing*, IEEE, 408-414.

[3]. Lee, J., Kao, H.A., and Yang, S., 2014. "Service innovation and smart analytics for industry 4.0 and big data environment". *Procedia CIRP*, 16, 3-8.

[4]. Krawczyk, B., & Woźniak, M. (2015). One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing*, 19(12), 3387-3400.

[5]. Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156.

[6]. Ma, C., Zhang, H.H., & Wang, X. (2014). Machine learning for big data analytics in plants. *Trends in Plant Science*, 19(12), 798-808.

[7]. Chen, X. W., & Lin, X. (2014). Big data deep learning: challenges and perspectives. *IEEE Access*, 2, 514-525.

[8]. Bassel, G.W., Gaudinier, A., Brady, S.M., Hennig, L., Rhee, S.Y., & De Smet, I. (2012). Systems analysis of plant functional, transcriptional, physical interaction, and metabolic networks. *The Plant Cell*, *24*(10), 3859-3875.

[9]. Yin, M.L. (2018). Implicit Learning for Efficient Maintenance Support. In: *Proceedings of the Annual Reliability and Maintainability Symposium*, IEEE, 1-6.

[10]. Cai, M., Pipattanasomporn, M., & Rahman, S. (2019). Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Applied Energy*, 236, 1078-1088.

[11]. Xie, J., Song, Z., Li, Y., & Ma, Z. (2018). Mobile big data analysis with machine learning. arXiv preprint arXiv:1808.00803.

[12]. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*;35(8), 1798–828.

[13]. Marcus, G. (2018). Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631.

[14]. Shi, S., Wang, Q., Xu, P., & Chu, X. (2016). Benchmarking state-of-the-art deep learning software tools. In: *Proceedings of the 7th International Conference on Cloud Computing and Big Data*, IEEE, 99-104.

[15]. Black, J.E., & Varma, A. (2012). Context sensitive help. U.S. Patent No. 8,151,192. Washington, DC: U.S. Patent and Trademark Office.

[16]. Chen, F., Ruiz, N., Choi, E., Epps, J., Khawaja, M. A., Taib, R., ... & Wang, Y. (2012). Multimodal behavior and interaction as indicators of cognitive load. *ACM Transactions on Interactive Intelligent Systems* , 2(4), 22.

[17]. Mortazavi, B., & AliShirvani, N. (2016). Multimodal user interface, enabling of assistive robots. *Bulletin de la Société Royale des Sciences de Liège*, 85, 446-453.

[18]. Turk, M. (2014). Multimodal interaction: A review. *Pattern Recognition Letters*, 36, 189-195.

[19]. Sonntag, D. (2015). Tutorial 3: intelligent user interfaces. In: *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, xxxii-xxxii.

[20]. Sonntag, D., Zillner, S., van der Smagt, P., & Lörincz, A. (2017). Overview of the CPS for smart factories project: deep learning, knowledge acquisition,

anomaly detection and intelligent user interfaces. *Industrial Internet of Things*, Springer, Cham, 487-504.

[21]. Zhou, J., Arshad, S.Z., Wang, X., Li, Z., Feng, D., & Chen, F. (2017). End-user development for interactive data analytics: Uncertainty, correlation and user confidence. *IEEE Transactions on Affective Computing*, 9(3), 383-395.

[22]. Zhou, J., & Chen, F. (2018). DecisionMind: revealing human cognition states in data analytics-driven decision making with a multimodal interface. *Journal on Multimodal User Interfaces*, 1-10.

[23]. Zhou, J., Arshad, S.Z., Yu, K., & Chen, F. (2016). Correlation for user confidence in predictive decision making. In: *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, 252-256.

[24]. Liu, X., Nielsen, P.S., Heller, A., & Gianniou, P. (2017). SciCloud: A scientific cloud and management platform for smart city data. In: *Proceedings of the 28th International Workshop on Database and Expert Systems Applications*, IEEE, 27-31.

[25]. Berwind, K., Voronov, A., Schneider, M., Bornschlegl, M., Engel, F., Kaufmann, M., & Hemmje, M. (2017). Big data reference model. *Big Data Analytics*, Chapman and Hall/CRC, 55-74.

[26]. Stankovic, J.A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1), 3-9.

[27]. Schleicher, J.M., Vögler, M., Dustdar, S., & Inzinger, C. (2016). Enabling a smart city application ecosystem: requirements and architectural aspects. *IEEE Internet Computing*, *20*(2), 58-65.

[28]. Ogiela, M.R., & Ogiela, L. (2018). Cognitive cryptography techniques for intelligent information management. *International Journal of Information Management*, *40*, 21-27.

[29]. Liu, B. (2019). An energy consumption control scheme based on radial basis function in wireless sensor networks. In: *Proceedings of the International Conference on Intelligent Transportation, Big Data & Smart City*, IEEE, 161-164.

[30]. Kendrick, C., Tan, K., Walker, K., & Yap, M. (2018). Towards real-time facial landmark detection in depth data using auxiliary information. *Symmetry*, 10(6), 1-18.

[31]. Yacchirema, D., Sarabia-Jácome, D., Palau, C.E., & Esteve, M. (2018). System for monitoring and supporting the treatment of sleep apnea using IoT and big data. *Pervasive and Mobile Computing*, 50, 25-40.

[32]. Li, W., Seneviratne, O., Patton, E., & Kagal, L. (2019). A semantic platform for developing data-intensive mobile apps. In: *Proceedings of the 13th International Conference on Semantic Computing*, IEEE, 71-78.

[33]. Sun, D., Lee, V.C., & Lu, Y. (2016). An intelligent data fusion framework for structural health monitoring. In: *Proceedings of the 11th Conference on Industrial Electronics and Applications*, IEEE, 49-54.

[34]. Brechtmann, F., Mertes, C., Matusevičiūtė, A., Yepez, V.A., Avsec, Ž., Herzog, M., ... & Gagneur, J. (2018). OUTRIDER: A statistical method for detecting aberrantly expressed genes in RNA sequencing data. *The American Journal of Human Genetics*, 103(6), 907-917.

[35]. Du Bois, E. (2013). A designerly methodology for software development. Doctoral dissertation, Delft University of Technology.

[36]. Horváth, I. (2019). A computational framework for procedural abduction done by smart cyber-physical systems. *Designs*, 3(1), 1-27.

[37]. Lees, B. (2013). Smart reasoning using case-based reasoning models. *Artificial Intelligence: Learning Model*s, Pearson Education: London, 1-3.

[38]. Kolodner, J. (2014). Case-based reasoning. Morgan Kaufmann.

[39]. Keane, M.T., & Kenny, E.M. (2019). How case based reasoning explained neural networks: An XAI survey of post-hoc explanation-by-example in ANN-CBR twins. arXiv preprint arXiv:1905.07186.

[40]. Li, O., Liu, H., Chen, C., & Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence,* 3530-3537.

[41]. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., & Kagal, L. (2018): Explaining explanations: an approach to evaluating interpretability of machine learning. arXiv preprint arXiv:1806.00069.

[42]. Moshirpour, M., Far, B., & Alhajj, R. (2018). Highlighting the importance of big data management and analysis for various applications. Springer.

[43]. Peng, H., Ma, S., & Spector, J.M. (2019). Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. *Foundations and Trends in Smart Learning*, Springer, Singapore, 171-176.

[44]. Bacca, J., & Segovia-Bedoya, D. (2019). An architecture for mobile-based assessment systems in smart learning environments. *Foundations and Trends in Smart Learning*, Springer, Singapore, 25-34.

[45]. Huang, R., Yang, J., & HU, Y. (2012). From digital to smart: The evolution and trends of learning environment. *Open Education Research*, 1(1), 75-84.

[46]. Akbari, N., & Modarressi, M. (2017). A high-performance network-on-chip topology for neuromorphic architectures. In: *Proceedings of the IEEE International Conference on Computational Science and Engineering*, IEEE, 2, 9-16.

[47]. Zhong, R.Y., Newman, S.T., Huang, G.Q., & Lan, S. (2016). Big data for supply chain management in the service and manufacturing sectors: Challenges, opportunities, and future perspectives. *Computers & Industrial Engineering*, 101, 572-591.

[48]. Suraj, M.V., Singh, N.K., & Tomar, D.S. (2018). Big data analytics of cyber attacks: A review. In: *Proceedings of the IEEE International Conference on System, Computation, Automation and Networking*, IEEE, 1-7.

[49]. Rizwan, A., Zoha, A., Zhang, R., Ahmad, W., Arshad, K., Ali, N.A., ... & Abbasi, Q. H. (2018). A review on the role of nano-communication in future healthcare systems: A big data analytics perspective. *IEEE Access*, 6, 41903-41920.

[50]. Srinivasan, S., & Kumari, T.T. (2018). Big data analytics tools a review. *International Journal of Engineering & Technology*, 7(2.33), 685-687.

[51]. Elshawi, R., Sakr, S., Talia, D., & Trunfio, P. (2018). Big data systems meet machine learning challenges: Towards big data science as a service. *Big Data Research*, 14, 1-11.

[52]. Imran, M., Ahamad, M.V., Haque, M., & Shoaib, M. (2018). Big data analytics tools and platform in big data landscape. *Handbook of Research on Pattern Engineering System Development for Big Data Analytics,* IGI Global, 80-89.

[53]. Saggi, M.K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing & Management*, 54(5), 758-790.

[54]. Gautam, C.S., & Pandey, P. (2019). A review of big data environment, tools and challenges. *Journal of Emerging Technologies and Innovative Research*, 6(6), 569-575.

[55]. Zhang, W., Ji, X., Yang, Y., Chen, J., Gao, Z., & Qiu, X. (2018). Data fusion method based on improved DS evidence theory. In: *Proceedings of the International Conference on Big Data and Smart Computing*, IEEE, 760-766.

[56]. Mönks, U., Dörksen, H., Lohweg, V., & Hübner, M. (2016). Information fusion of conflicting input data. *Sensors*, 16(11), 1798-1835.

[57]. Ban, Y., & Jacob, A. (2013). Object-based fusion of multitemporal multiangle ENVISAT ASAR and HJ-1B multispectral data for urban land-cover mapping. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4), 1998-2006.

[58]. Byun, Y. (2014). A texture-based fusion scheme to integrate highresolution satellite SAR and optical images. *Remote Sensing Letters*, 2014, 5(2), 103-111.

[59]. Toussaint, G.T. (1978). The use of context in pattern recognition. *Pattern Recognition*, 10(3), 189-204.

[60]. Garain, U. (2016). A stochastic approach for finding optimal context in a contextual pattern analysis task. *IEEE Intelligent Systems*, 31(2), 21-28.

[61]. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness, *Lecture Notes on Computer Science*, 304-307.

[62]. Berardinelli, L., Cortellessa, V., & Di Marco, A. (2010). Performance modeling and analysis of context-aware mobile software systems. *Fundamental Approaches to Software Engineering*, Springer, Berlin Heidelberg, 353-367.

[63]. Li, Y. (2019). Utilizing dynamic context semantics in smart behavior of informing cyber-physical systems. Doctoral dissertation, Delft University of Technology.

[64]. Adam, E.V. (1993). Fighter cockpits of the future. In: *Proceedings of the 12th IEEE Conference on Digital Avionics Systems*, 318-323.

[65]. Endsley, M.R. (1995) Toward a theory of situation awareness in dynamic systems. *Human Factors and Ergonomic Society*, 37(1), 32-64.

[66]. Loia, V., D'Aniello, G., Gaeta, A., & Orciuoli, F. (2016). Enforcing situation awareness with granular computing: a systematic overview and new perspectives. *Granular Computing*, 1(2), 127-143.

[67]. Nwiabu, N. (2014). Situation awareness for violence prevention on campuses. In: *Proceedings of the International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, IEEE, 90-96.

[68]. Orji, F., & Igiri, C.G. (2017). Big data system of systems for situation awareness in safety and security. *West African Journal of Industrial & Academic Research*, 18(1), 61-67.

[69]. Chan, L., Morgan, I., Simon, H., Alshabanat, F., Ober, D., Gentry, J., ... & Cao, R. (2019). Survey of AI in cybersecurity for information technology management. In: *Proceedings of the Technology & Engineering Management Conference*, IEEE, 1-8.

[70]. Bergmann, R., Kolodner, J., & Plaza, E. (2005). Representation in case-based reasoning. *The Knowledge Engineering Review*, 20(3), 209-213.

[71]. Schmidt, G. (1998). Case-based reasoning for production scheduling. *International Journal of Production Economics*, 56, 537-546.

[72]. Alsukhni, M. & Zhu, Y. (2012). Interactive visualization of the social network of research collaborations. In: *Proceedings of the 13th IEEE International Conference on Information Reuse & Integration,* Las Vegas, NV, 247-254.

[73]. Matin, A.I., Jahan, S. & Huq, M.R. (2014). Community recommendation in social network using strong friends and quasi-clique approach. In: *Proceedings of the International Conference on Electrical and Computer Engineering, Dhaka*, 453-456.

[74]. Yadav, A.K., & Shah, A.I. (2016). A Study on Requirements and Characterization of Recommender System for Social Network. International Journal of Computer Science and Mobile Computing, 5(7), 457-461.

[75]. Archenaa, J., & Anita, E.M. (2017). Health recommender system using big data analytics. *Journal of Management Science and Business Intelligence*, 17-24.

[76]. Tripathi, P., Agarwal, R., & Vashishtha, T. (2016). Review of job recommender system using big data analytics. In: *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development*, IEEE, 3773-3777.

[77]. Isaacson, S.A., Anderson, E.W., & Wipfel, R. (2013). *U.S. Patent No. 8,352,935*. Washington, DC: U.S. Patent and Trademark Office.

[78]. Cannata, J.B. (1999). Time is running out for customized software: resolving the goods versus service controversy for year 2000 contractual disputes. *Cardozo L. Rev.*, 21, 283.

[79]. Punch, K.F. (2005). Introduction to social research. London, Sage.

[80]. Rejnö, Å., Danielson, E., & Berg, L. (2017). Combining qualitative and quantitative content analysis: a way to utilize different types of data in research of ethical problems in acute stroke care. Sage Publications Ltd.

[81]. Hou, Y.A. (2018). Avoiding the gap of college students' internship expectations and perceptions—a case study in Taiwan. *Open Journal of Nursing*, 8(08), 531.

[82]. Dai, H.N., Wong, R.C.W., Wang, H., Zheng, Z., & Vasilakos, A.V. (2019). Big data analytics for large-scale wireless networks: Challenges and opportunities. *ACM Computing Surveys*, 52(5), 1-36.

[83]. Guerra, J., Pucha, H., Glider, J.S., Belluomini, W., & Rangaswami, R. (2011). Cost effective storage using extent-based dynamic tiering. In: *Proceedings of the 9th USENIX Conference on File and Storage Technologies*, USENIX Association, 11, 20-20.

[84]. Blurton, C. (1999). New directions of ICT-Use in education. Retrieved from: www.unesco.org/education/ lwf/dl/edict.pdf .

[85]. Marinac, I., Napoli, M.B., Fitzgerald, A., McLachlan, K., Matkovic, D., & Trotic, R. (2018). The role of web-based implant platform in raising of the implantee's quality of life. *Journal of Hearing Science*, 8(2).

[86]. Santra, A.K., & Nagarajan, S. (2012). Authentication algorithm for intrusion detection in wireless networks. *International Journal of Computer Science Issues*, 9(2), 551.

[87]. Hawas, A.R., El-Khobby, H.A., Abd-Elnaby, M., & El-Samie, F.E.A. (2019). Gait identification by convolutional neural networks and optical flow. *Multimedia Tools and Applications*, 78(18), 25873-25888.

[88]. Ali, M.L., Monaco, J.V., Tappert, C.C., & Qiu, M. (2017). Keystroke biometric systems for user authentication. *Journal of Signal Processing Systems*, 86(2-3), 175-190.

[89]. Kasuya, M., & Sakiyama, K. (2017). Improved EM side-channel authentication using profile-based XOR model. In: *Proceedings of the International Workshop on Information Security Applications*, Springer, Cham, 173-183.

[90]. Rao, Y.S., Sukonkina, Y., Bhagwati, C., & Singh, U.K. (2008). Fingerprint based authentication application using visual cryptography methods (improved ID card). In: *Proceedings of the IEEE Region 10 Conference*, IEEE, 1-5.

[91]. Nayak, A., & Bansode, R. (2016). Analysis of knowledge based authentication system using persuasive cued click points. *Procedia Computer Science*, 79, 553-560.

[92]. Joyce, R., & Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2), 168–176.

[93]. Town, P., & Thabtah, F. (2019). Data analytics tools: a user perspective. *Journal of Information & Knowledge Management*, 18(01), 1-16.

[94]. Meincke, D. (2018). Can Robots Replace Librarians? Experiences Using a Chat Bot to Respond to IM Questions.

[95]. Kerr, H.T., Ehinger, M.H., Wachter, J.W., & Hebble, T.L. (1986). Process monitoring for reprocessing plant safeguards: a summary review. Oak Ridge National Lab.

[96]. Leem, C.S. (1996). An automated fault diagnosis for manufacturing process monitoring and control. *International Journal of Modelling and Simulation*, 16(4), 200-209.

[97]. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

[98]. Amini, M., & Chang, S. (2018). A review of machine learning approaches for high dimensional process monitoring. In: *Proceedings of the 2018 Industrial and Systems Engineering Research Conference*, 1-6.

[99]. Madhikermi, M., Buda, A., Dave, B., & Främling, K. (2017). Key data quality pitfalls for condition based maintenance. In: *Proceedings of the 2nd International Conference on System Reliability and Safety*, IEEE, 474-480.

[100]. Wang, L., Wang, G., & Alexander, C.A. (2015). Big data and visualization: methods, challenges and technology progress. *Digital Technologies*, 1(1), 33-38.

[101]. Khan, A.M., Gonçalves, D., Leão, D.C., Puig, A.P., & Isenberg, T. (2017). Towards an adaptive framework for real-time visualization of streaming big data. *EuroVis (Posters)*, 5-7.

[102]. Canning, C.A., Loe, A., Cockett, K.J., Gagnon, P., & Zary, N. (2017). Data driven quality improvement of health professions education: design and development of CLUE–an interactive curriculum data visualization tool. IOS Press.

[103]. Zhong, Y., Wang, S., Li, R., Yun, W., & Song, G. (2017). Design and implementation of high-performance GIS dynamic objects rendering engine. *AGU Fall Meeting Abstracts*.

[104]. Moyne, J., Samantaray, J., & Armacost, M. (2016). Big data capabilities applied to semiconductor manufacturing advanced process control. *IEEE Transactions on Semiconductor Manufacturing*, 29(4), 283-291.

[105]. Mitchell, H.B. (2007). Multi-sensor data fusion: An introduction. *Springer Science & Business Media*.

[106]. Paradis, L., & Han, Q. (2007). A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management*, 15(2), 171-190.

[107]. Khaleghi, B., Khamis, A., Karray, F.O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1), 28-44.

[108]. Bleiholder, J., & Naumann, F. (2009). Data fusion. *ACM Computing Surveys*, 41(1), 1-41.

[109]. Boulkaboul, S., & Djenouri, D. (2020). DFIOT: Data fusion for internet of things. *Journal of Network and Systems Management*, 1-25.

[110]. Schmitt, M., & Zhu, X.X. (2016). Data fusion and remote sensing: An ever-growing relationship. *IEEE Geoscience and Remote Sensing Magazine*, 4(4), 6-23.

[111]. Lyubchyk, L. M. (2014). Optimal information fusion in stochastic unknown input observer network. *Advanced Data Acquisition and Intelligent Data Processing, River Publishers*, Denmark, 127-158.

[112]. Rivera-Castillo, J., Lindner, L., …, & Basaca-Preciado, L.C. (2019). Applying optoelectronic devices fusion in machine vision. *Natural Language Processing: Concepts, Methodologies, Tools, and Applications*, 184-213.

[113]. Castanedo, F. (2013). A review of data fusion techniques. *The Scientific World Journal*, 2013, 1-19.

[114]. Durrant-Whyte, H., & Henderson, T.C. (2016). Multisensor data fusion. In *Springer Handbook of Robotics,* Springer, Cham, 867-896.

[115]. Zhao, Q., Ma, G., Wang, Q., Yang, T., Liu, M., Gao, W., ... & Pepe, A. (2019). Generation of long-term InSAR ground displacement time-series through a novel multi-sensor data merging technique: The case study of the Shanghai coastal area. *Journal of Photogrammetry and Remote Sensing*, 154, 10-27.

[116]. Outamazirt, F., Li, F., Yan, L., & Nemra, A. (2014). Autonomous navigation system using a fuzzy adaptive nonlinear H∞ filter. *Sensors*, 14(9), 17600-17620.

[117]. Hall, D.L., & Llinas, J. (2017). Multisensor data fusion. *Handbook of Multisensor Data Fusion*, CRC Press, 21-34.

[118]. Hall, D.L., Chong, C.-Y., LLinas, J., & Liggins II, M. (2017). Distributed data fusion for network-centric operations. CRC Press.

[119]. Gite, S., & Agrawal, H. (2016). On context awareness for multisensor data fusion in IoT. In: *Proceedings of the 2nd International Conference on Computer and Communication Technologies*, Springer, New Delhi, 85-93.

[120]. Barde, A., & Jain, S. (2018). A survey of multi-sensor data fusion in wireless sensor networks. In: *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies*, 26-27.

[121]. Kumar, M., Garg, D.P., & Zachery, R.A. (2006). A generalized approach for inconsistency detection in data fusion from multiple sensors. In: *Proceedings of the American Control Conference*, IEEE, 2078-2083.

[122]. Smets, P. (2007). Analyzing the combination of conflicting belief functions. *Information fusion*, 8(4), 387-412.

[123]. Gao, L. (2017). A discriminative analysis framework for multi-modal information fusion. Doctoral dissertation, Ryerson University.

[124]. Alam, F., Mehmood, R., Katib, I., Albogami, N.N., & Albeshri, A. (2017). Data fusion and IoT for smart ubiquitous environments: A survey. *IEEE Access*, 5, 9533-9554.

[125]. Gangadhar, S. (2017). Sensor fusion framework and simulation on a Turtlebot robotic vehicle. Doctoral dissertation, The University of North Carolina at Charlotte.

[126]. Smith, D., & Singh, S. (2006). Approaches to multisensor data fusion in target tracking: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(12), 1696-1710.

[127]. Khaleghi, B., Razavi, S.N., Khamis, A., Karray, F.O., & Kamel, M. (2009). Multisensor data fusion: Antecedents and directions. In: *Proceedings of the 3rd International Conference on Signals, Circuits and Systems*, IEEE, 1-6.

[128]. Khaleghi, B. (2012). Distributed random set theoretic soft/hard data fusion. Doctoral dissertation, The University of Waterloo.

[129]. Joshi, R., & Sanderson, A.C. (1999). Multisensor fusion: A minimal representation framework. World Scientific.

[130]. Dong, X.L., Berti-Equille, L., & Srivastava, D. (2009). Truth discovery and copying detection in a dynamic world. In: *Proceedings of the Very Large Data Base Endowment*, 2(1), 562-573.

[131]. Milenova, B.L., & Campos, M.M. (2005). Mining high-dimensional data for information fusion: a database-centric approach. In: *Proceedings of the 7th International Conference on Information Fusion*, IEEE, 1, 638-645.

[132]. Abbas, M.A., Benblidia, N., & Bachari, N.E.-I. (2019). A multi-level fusion approach for climate variation study using multi-source data - case study: Algeria. In: *Proceedings of the 16th International Conference on Computer Systems and Applications*, IEEE, 1-7.

[133]. Arunkumar, A., Surendran, D., & Sreya, S. (2019). Data fusion and machine learning in medical diagnosis: A bird eye view. *Journal of Computational and Theoretical Nanoscience*, 16(12), 5127-5133.

[134]. Dautov, R., Distefano, S., & Buyya, R. (2019). Hierarchical data fusion for smart healthcare. *Journal of Big Data*, 6(1), 1-23.

[135]. Li, X., & Guo, Z. (2018). Multi-source information fusion model of traffic lifeline based on improved DS evidence theory. In: *Proceedings of the 26th International Conference on Geoinformatics*, IEEE, 1-6.

[136]. Zhang, L., Xie, Y., Xidao, L., & Zhang, X. (2018). Multi-source heterogeneous data fusion. In: *Proceedings of the International Conference on Artificial Intelligence and Big Data, IEEE*, 47-51.

[137]. Slaughter, L. M. (2019). *A* modular semi-supervised sensor fusion method for inferring real time vacancy in buildings. Doctoral dissertation, University of California.

[138]. Srivastava, S.K., & Roy, S.N. (2018). Recommendation system: A potential tool for achieving pervasive health care. *Next-Generation Mobile and Pervasive Healthcare Solutions.* IGI Global, 111-127.

[139]. Lops, P., Jannach, D., Musto, C., Bogers, T., & Koolen, M. (2019). Trends in content-based recommendation. *User Modeling and User-Adapted Interaction*, 29(2), 239-249.

[140]. Fu, W., Liu, J., & Lai, Y. (2018). Collaborative filtering recommendation algorithm towards intelligent community. *Discrete & Continuous Dynamical Systems-S*, 12(4&5), 811-822.

[141]. Jokela, T., Iivari, N., Matero, J., & Karukka, M. (2003). The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In: *Proceedings of the Latin American Conference on Human-Computer Interaction*, ACM, 53-60.

[142]. Syukri, A.F., Okamoto, E., & Mambo, M. (1998). A user identification system using signature written with mouse. In: *Proceedings of the Australasian Conference on Information Security and Privacy*, Springer, Berlin, Heidelberg, 403-414.

[143]. Eze, P.C., Achebe, P.N., Jeremiah, L., & Ageh, T.A. (2018). Anti-Theft System for Car Security using RFID.

[144]. Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24(11), 770-772.

[145]. Curran, J., & Curran, K. (2019). Biometric authentication techniques in online learning environments. *Biometric Authentication in Online Learning Environments*, IGI Global, 266-278.

[146]. Reese, K., Smith, T., Dutson, J., Armknecht, J., Cameron, J., & Seamons, K. (2019). A usability study of five two-factor authentication methods. In: *Proceedings of the 15th Symposium on Usable Privacy and Security*, 357-370.

[147]. Li, H., Dai, Y., Tian, L., & Yang, H. (2009). Identity-based authentication for cloud computing. In: *Proceedings of the IEEE International Conference on Cloud Computing*, Springer, Berlin, Heidelberg, 157-166.

[148]. Yao, Z., Li, B., & Liu, S. (2011). Role based collaboration authorizing by using ontology. *Chinese Journal of Electronics*, 20(3), 389-394.

[149]. Jablon, D.P. (1996). Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, 26(5), 5-26.

[150]. Lin, C. L., Sun, H.M., & Hwang, T. (2001). Attacks and solutions on strong-password authentication. *IEICE Transactions on Communications*, 84(9), 2622-2627.

[151]. Bhattacharyya, D., Ranjan, R., Alisherov, F., & Choi, M. (2009). Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology*, 2(3), 13-28.

[152]. Das, M.L. (2009). Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 8(3), 1086-1090.

[153]. Sako, K., Yonezawa, S., & Teranishi, I. (2005). Anonymous authentication: For privacy and security. *NEC Journal of Advanced Technology*, 2(1), 79-83.

[154]. Yao, H. (2011). Role-based authentication protocol. *Security and Communication Networks*, 4(4), 460-470.

[155]. Pomputius, A.F. (2018). A review of two-factor authentication: suggested security effort moves to mandatory. *Medical Reference Services Quarterly*, 37(4), 397-402.

[156]. Erden, Z., Erkmen, A.M., & Erden, A. (1998). A petri net-based design network with applications to mechatronic systems. *Journal of Integrated Design and Process Science*, 2, 32-48.

Chapter 5

Research cycle 4:
Implementation and validation of the functions of the demonstrative smart data analytics toolbox

5.1. Introduction

5.1.1. Objectives of the fourth research cycle

The objective of the fourth research cycle was to implement the representative computational functions of the demonstrative data analytics toolbox, and test the proposed concepts and the implementation of algorithms and data constructs. To this end, the first part of the work was dedicated to the detailing of the algorithms in executable forms and to the implementation of each function and module of the demonstrative SDATB. The starting point for this research cycle was the functional decomposition presented in Chapter 4. The implemented functions and modules were tested in a specific application case. Using the reasoning with consequences principle, we validated the feasibility of the proposed functions. The application scenario considered enhancement of white goods by product designers using MoLD. We determined the expected inputs and outputs related to the application scenario. A particular type of white good, a connected washing machine, was the subject of the design scenario.

5.1.2. Methodology applied in the fourth research cycle

Like the previous one, this research cycle was framed according to the structure of the DIR methodology, with an ODR flavor. It included three phases: (i) explorative, (ii) constructive, and (iii) confirmative. The specific tasks completed in the explorative phase were the following: (i) determining and detailing the algorithms to be used for the implementation of the demonstrative SDATB, (ii) reviewing existing solutions that enable the implementation of data structures and algorithms for each function, (iii) collecting information about the fundamentals that support designing the needed algorithms, and (iv) investigating the logical and computational techniques for the prototype-level implementation in relation to the application case. The constructive phase focused on the software-level implementation of the modules and the algorithms. Finally, the confirmative phase validated the functionality of the implemented modules. This validation was methodologically framed according to ODR.

5.2. Algorithm-level specification of the demonstrative smart data analytics toolbox

The chosen representative functions imply one or more interoperable algorithms that are needed for the software-level implementation of the SDATB. The novel and significant algorithms are specified in this section. In developing the algorithms, we considered the reference application case of white goods enhancement based on MoLD. Below, we explain the algorithms and detail their contents.

5.2.1. Algorithm-level specification of the merging middle-of-life data streams recommendation module

The recommendation module for merging MoLD-Ss is reasonably novel. Table 5.1 lists the algorithms needed for the realization of this module. Three algorithms are needed for the sub-function $F_{SB1,1}$. The first (algorithm A_1) is responsible for requesting from the designer the list of sensors to be analyzed by the SDATB. The second (algorithm A_2) requests access to data streams and their locations. The third (algorithm A_3) is responsible for acquiring MoLD-Ss from remote storage (for example, a cloud environment) and moving them to the SDATB and its local storage. For sub-function $F_{SB1,2}$, two algorithms are needed. One is responsible for providing means for visualizations (plotting) to comprehend data despite their raw format (algorithm A_4). The second one (algorithm A_5) is responsible for the normalization of MoLD-Ss so the data streams can be properly used for further analyses. This algorithm is needed to remove anomalies that might complicate the analysis, such as by (i) deleting data (e.g. removing correlated time series), (ii) inserting more information (e.g. applying one hot encoding for categorical features), or (iii) updating existing information (e.g. clipping outliers).

For sub-function $F_{SB1,3}$, four algorithms are needed. Algorithm A_6 is responsible for processing normalized single MoLD-Ss time series using a statistical model. This is needed to generate length-invariant representations of MoLD-Ss to reduce computation costs in the upcoming steps. Algorithm A_7 calculates or estimates the importance of the sensors to analyze. This is only needed when a large number of sensors are to be merged and analyzed. By considering a lower number of relevant data streams, the interpretation of predictions is improved. The outcomes of this steps are used in Algorithm A_8, which is an algorithm run in Matlab. This algorithm has been constructed to merge data streams that are obtained from various sensors but which are captured in the same time frame. The developed algorithms are for semantic fusion based on estimating anomalies and performing similar descriptor searches in the database. Algorithm A_8 considers the weights allocated to sensors and selects only those with the highest weight values for merging. This means we order the sensors according to their estimated fusion weights and consider a portion of the most relevant MoLD-Ss in the merging. Algorithm A_9 processes the MoLD-S jointly and embeds information into a new latent space (or representation). In such a space, a distance reflects the degree of semantic similarity. The behavior descriptor is sensor independent and describes the behavior independently from the source.

To realize sub-function $F_{SB1,4}$, six algorithms are needed. Algorithm A_{10} is responsible

Table 5.1. Specified algorithms for merging middle-of-life data streams (MoLD-Ss) recommendation module

Sub-function code	Algorithm code	Specification of the algorithm
$F_{SB1,1}$	A_1	Request list of sensors
	A_2	Request a subset of devices supporting provided sensors
	A_3	Fetch MoLD-Ss to the SDATB
$F_{SB1,2}$	A_4	Plot sensors' data streams as time series for selected data streams
	A_5	Apply time series normalization for each MoLD-S
$F_{SB1,3}$	A_6	Process single stream time series with statistical model
	A_7	Estimate sensors' importance
	A_8	Merge MoLD-Ss based on fusion weights
	A_9	Estimate behavior descriptor based on merged MoLD-Ss representation
$F_{SB1,4}$	A_{10}	Estimate probability of anomaly
	A_{11}	Perform similar descriptor search in database
	A_{12}	Calculate distance between anomalies
	A_{13}	Rank anomaly descriptors by their distance from a requested one
	A_{14}	Retrieve relevant anomalies based on ranking as well as their corresponding sensors
	A_{15}	Merge relevant anomalies into an action plan (recommendation)
$F_{SB1,5}$	A_{16}	Select recommendation message template
	A_{17}	Convert individual anomalies into recommendation message component
	A_{18}	Convert the action plan into recommendation message component
	A_{19}	Order recommendation message components
	A_{20}	Integrate recommendation message components according to template

for estimating anomalies probability. It is a preliminary step to a more thorough search through the knowledge database containing a list of pre-programmed anomalies. It consists of calculating the distance to knowledge anomalies in the database. Algorithm A_{11} gathers similar past anomalies from the database. It performs a search for similar descriptors, iterating through the pairs of the detected anomalies and the past ones.

Algorithm A_{12} calculates the pairwise distance between the detected and the past anomalies. These anomalies are ranked via Algorithm A_{13} and then retrieved using

Algorithm A_{14}. Algorithm A_{15} executes the semantic merging of the retrieved anomalies and generates a recommendation, which contains an action plan detailing what needs to be done with the product.

Realizing sub-function $F_{SB1,5}$ requires five algorithms. Algorithm A_{16} selects the template for the recommendation message from the database. Algorithms A_{17} and A_{18} successively convert the detected anomalies and the action plan into components of the recommendation message. Algorithm A_{19} executes the ordering of the appearance of individual anomalies and includes an action plan in the recommendation message. Algorithm A_{20} integrates the ordered components of the message into the template to provide the recommendation message to be presented to the designer.

5.2.2. Algorithm-level specification of the task-relevant data analytics tools recommendation module

The algorithms required for the realization of the task-relevant data analytics tools recommendation module are listed in Table 5.2. Due to the novelty of this main function, several new algorithms had to be constructed. For sub-function $F_{SA1,1}$, Algorithm A_{21} recognizes the task specified by the designer. This algorithm calculates the minimum distance between the DT given by designer and DTs saved in the database of the SDATB, and it determines the closest DT and DST considering the saved DTs. For sub-function $F_{SA1,2}$, Algorithm A_{22} is used to characterize DSTs based on DS, DC, and DO and to match the data analytics tools saved in the database of the SDATB. Three criteria have been considered: (i) data source of the dataset to be analyzed (DS), (ii) data category of the dataset (DC), and (iii) output of data analytics (DO). These elements help creating a matching matrix between DATs and DTs.

The procedure includes the conversion of the textual description of DSTs to a vector of words, and the comparison of it with constant character vectors representing relationships between DST_x and DT_x. To realize $F_{SA1,3}$, three algorithms were

Table 5.2. Algorithms specified for the task-relevant data analytics tools recommendation module

Sub-function code	Algorithm code	Specification of the algorithm
$F_{SA1,1}$	A_{21}	Retrieve DST with minimum distance to DT given by designer
$F_{SA1,2}$	A_{22}	Build DSTs vectors
$F_{SA1,3}$	A_{23}	Build vector DAT
	A_{24}	Calculate distance between DSTs and DAT vectors
	A_{25}	Sorting DATs
	A_{26}	Retrieve DAT vectors most similar to DST vector
$F_{SA1,4}$	A_{27}	Rank DATs
$F_{SA1,5}$	A_{28}	Retrieve best finite set of DATs

considered. Algorithm A_{23} builds a vector for all DATs in the SDATB that includes the DS, DC, and DO. Algorithm A_{24} calculates the distance between the vectors DST and DATs. Algorithm A_{25} sorts DATs based on their distance to the DST. Algorithm A_{26} retrieves all DATs with equal minimum distance to the DST and their corresponding weights. For sub-function $F_{SA1,4}$, one algorithm (Algorithm A_{27}) is considered, which ranks the tools with minimum distance to the DST using the matrix of their corresponding weights.

Every tool has different weights based on the criteria. Therefore, we first generate a vector with all weights for each tool, then calculate the sum of weights. Once we have the values of all sums of the weights, we order them to identify those with maximum sum. This ordering also sorts the tools, and tool with the maximum sum is the tool to select. Finally, sub-function $F_{SA1,5}$ uses one algorithm (Algorithm A_{28}) to select the tools with the maximum sum of weights and presents them to the designer as a set of the best-matching tools.

5.2.3. Algorithm-level specification of the smart user identification module

The algorithms needed for the smart user identification module are presented in Table 5.3. The realization of this module is application independent, since no specifications are needed from the user. Neither is it directly related to the data processing to be done by the toolbox. As mentioned earlier, several face recognition approaches are described in the literature. We followed one that can be referred to as a standard approach. The first step of face recognition is acquisition of the input image, followed by face detection, which is performed to determine whether a face appears in the captured image. The process concludes with locating the position of the face in the image. This face image is

Table 5.3. Specified algorithms for the smart user identification module

Sub-function code	Algorithm code	Specification of the algorithm
$F_{SI1,1}$	A_{29}	Acquire new image from camera
	A_{30}	Convert color to grayscale
	A_{31}	Detect face in image
	A_{32}	Normalize face image (size and orientation)
	A_{33}	Capture face
$F_{SI1,2}$	A_{34}	Face features extraction
	A_{35}	Face features matching
	A_{36}	User identity validation
$F_{SI1,3}$	A_{37}	Generate attendance file
$F_{SI1,4}$	A_{38}	Generate database
$F_{SI1,5}$	A_{39}	Reset attendance file
	A_{40}	Reset database

then preprocessed using normalization techniques to remove illuminations, shades, and lighting effects without affecting face features. The features of the normalized image are extracted using a subspace framework. In the end, the extracted features are trained to a neural network using a subspace classifier to obtain the identified, recognized image [1] [2]. Projecting these steps to our smart user identification module, we identify a need for the following computational algorithms.

For sub-function $F_{SI1,1}$, five algorithms (A_{29} to A_{33}) are needed to detect and extract the designer's (user's) face image. To complete sub-function $F_{SI1,2}$, three algorithms (A_{34} to A_{36}) are needed. Algorithm A_{34} is the actual face recognition algorithm. Algorithm A_{35} collects face samples to train the face recognition algorithm using previously captured face samples. Algorithm A_{36} matches the images with the pattern invoked by the training data. Sub-function $F_{SI1,3}$ requires one algorithm (A_{37}), which generates the attendance file (document) for the designer, including the time and date of access to the SDATB. Sub-function $F_{SI1,4}$ also needs one algorithm (A_{38}) to generate the database of the designer with the images captured during identifications. Finally, sub-function $F_{SI1,5}$ requires two algorithms (A_{39} and A_{40}) to reset the attendance file and the database, if needed by the designer.

5.3. Fundamentals of the computational implementation of the demonstrative modules

5.3.1. Fundamentals of the implementation of the merging middle-of-life data streams module

MSDM technologies are widely used in many application fields [3]. They focus on sensing different signals from various sensors, analyzing specific parameters, and integrating them as output information [4]. They are considered very important in the process of decision-making, since data extracted from a single sensor is not sufficient for making a decision [5]. Several theories have been presented in the literature to address multi-sensor data fusion, such as (i) rough sets theory [6], (ii) evidence theory [7], (iii) fuzzy sets theory [8], (iv) evidential reasoning [9], (v) statistical learning theory [10], and others [11] - [16]. Rough sets theory provides systematic representation to data imprecision and granularity [17]. It does not require prior knowledge and represents imprecise data solely based on its internal structure [18].

Evidence theory is flexible and effective in modeling uncertainty and imprecision regardless of prior information [19], but counterintuitive results can be generated when fusing highly conflicting evidence [20]. Fuzzy sets theory focuses on intuitive reasoning by considering human subjectivity and imprecision [21]. Evidential reasoning deals with problems containing qualitative and quantitative criteria under multiple uncertainties, including ignorance and randomness [22]. Statistical learning theory is a framework for machine learning merging statistics and functional analysis [23]. It focuses on finding a predictive function for a task or a problem [24].

The developed module focuses on merging multiple streams, detecting anomalies, and offering recommendations to the designer based on them. Consequently, statistical learning theory is used because of the opportunities it offers for anomaly detection in the case of multi-sensor data fusion [25-29]. Some researchers consider it to be “one of the most beautifully developed branches of artificial intelligence” [30]. The objective of the learning is prediction [31]. The theory represents the process of inferring general rules by observing examples [30]. Mathematically speaking, the objective of statistical learning, also called statistical machine learning, is prediction of an unobserved output y based on an observed input x [32]. In machine learning problems, the objective is not to find the function that best fits the previously observed data but to find the one that will accurately predict output from future input [33]. The most common techniques of statistical machine learning are (i) ANNs, (ii) learning with hidden variables, (iii) instance-based learning, and (iv) kernel machines.

ANNs are adapted for fusing data from multiple sensors. They are well suited for the combination of inputs from completely different sources [34]. Accordingly, we selected them to achieve the desired function. ANNs are a standard paradigm for recognizing patterns [35]. They are very well-suited for sensor, information, and decision-fusion scenarios formulation [36]. Neural networks are being used for image classification, single-object localization, object detection, and semantic segmentation [37]. They are also used for continuous vector representations of words [38], unsupervised clustering, dimensionality reduction, data visualization [139], and semi-supervised classification with the help of an auto encoder model [40]. Moreover, building a recommendation can be done using neural network-based solutions [41] [42]. Such solutions solve issues related to information overload [43], decision-making in multiple-choice scenarios [44], and feature representation learning without prior knowledge [45].

Statistical learning theory requires a dataset of prior evidence. To find a neural network that predicts a required output, an optimization process based on gradient descent has been proposed [46]. The most commonly used one in the literature is Adam [47]. Related to neural networks, the exponential growth of datasets has implications for the boundaries of the most commonly used optimization techniques. The difficulty is achieving a linear reduction in function approximation. The observed exponential growth is challenging and makes it difficult for neural networks to predict outputs [48].

Many efforts have been dedicated to convergence challenges related to neural networks. To guarantee linear convergence, a group of researchers developed a reduced set of neural networks with strong requirements for datasets [49]. Other researchers analyzed the optimization of variable-depth neural networks to achieve exponential scaling of convergence with increasing depth [50]. In practice, multilayered neural networks can still be efficiently trained with stochastic gradient descent. Given a fixed amount of data, generalization can be improved by changing the architecture of the neural network.

Methods with strong performance have been developed to automatically find the needed neural network architecture and to obtain good models for various pattern-recognition tasks [51]. The resulting architectures are novel and surpass human-invented ones. The dataset preprocessing step also plays an important role. This step includes whitening, decorrelation, internal covariant, and shift reduction, which can reduce the time needed for the training process by a factor of 14 (14 times faster) [52]. Needless to say, several methods of preprocessing are described in the literature. The preprocessing approach we prefer involves normalizing single data streams over time. It is considered to be the key preprocessing step [53] and reduces unwanted variations in data streams [54]. If $x_{t,i}$ is a

MoLD-Ss generated by a sensor of order number *i*th, recorded at time *t*, then the normalized MoLD-Ss, represented as x'_t, is calculated by the following equation:

$$x'_t = \frac{x_t - \mu_x}{\sigma_x + \epsilon}, \tag{5.1}$$

where μ_x is the vector of mean, estimated per each stream across time

$$\mu_x = \frac{1}{T} \sum_{i=1}^{T} x_{t,i}, \tag{5.2}$$

σ_x is the vector of standard deviations, estimated for each stream across time

$$\sigma_x = \frac{1}{T} - 1 \sum_{i=1}^{T} \left(x_{t,i} - \mu_x\right)^2, \tag{5.3}$$

and T is the number of time steps in the data stream. Note that an arbitrarily small positive value, ϵ, is introduced to avoid division by 0.

The step following normalization is time series analysis. This step requires an effective model independent of the size of the input sequence. Other studies have applied nonrecurrent models for generation and analysis of high-frequency data [55]. In our case, we considered a length-limited data stream of T time steps, and we applied the algorithm called “sliding window.” Its objective is to transform a long sequence by breaking it into short time frames with potentially overlapping boundaries [56]. Applying this algorithm results in analyzing portions of long time series. Symbolically, the statistical model operates as follows:

$$h = f(x), \tag{5.4}$$

where $x = x^T_i = 0$ is the raw data stream, f is the neural network encoder, and h is the fixed length representation. To guarantee that semantics are captured in MoLD-Ss at merging, we use an unsupervised learning approach to learn the importance of sensors based on weighting the important features from data streams. Computationally, this happens based on the attention layer [57]. The attention layer is typically used for audio, video, and text fusion [58]. In our case, we applied it for weighting important latent features in input embedding [59].

Let us define $H \in R^{M,L}$ as an intermediate representation of time series of each sensor present in the multimodal data stream (coming from multiple sensor data streams) and $W \in R^{L,L}$ and $v \in R^L$ as parameters of attention. As used in the neural network, the normalized exponential function, called *softmax*, takes a vector of K real numbers as input and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. Thus, the reweighting of the sensors embedding is as follows:

$$A(H \mid W, v) = softmax(v \times tanh(H \times W)'), \tag{5.5}$$

where *softmax* is defined as follows:

$$u = g(v) \quad ; \quad u_i = \frac{exp(v_i)}{\sum_{i=1}^{N} \exp(v_j)}\,. \tag{5.6}$$

The range of computation is from 1 to N. As a result, the weights of each sensor data stream are

$$w_i = f\big(g(X_i)\big)\,, \tag{5.7}$$

where X'_i is the i_{th} single data stream after preprocessing being passed through the sensor encoder g and attention estimator f. To merge MoLD-Ss, we reweighted each stream's latent representation and apply a behavior encoder:

$$L_i = g(X_i) \quad ; \quad h = z(L \cdot w), \tag{5.8}$$

where L is a merged stream (representation) from a sensor, h is a behavior descriptor of the whole stream, and z is the function allowing calculation of the behavior descriptor of all streams L with corresponding weights w.

A behavior encoder model needs an objective for optimization. The triplet network model has been introduced to learn useful representations from distance comparisons [60]. The model is capable of providing rich vector representations for classification datasets. Let us define (u,v,w) as a triplet, where u represents the behavior descriptor used as an anchor, and v and w are to be compared with u by their distance. The triplet loss function minimizes $d(u,v)$ and maximizes $d(u,w)$. The model constrains the behavior descriptor to cluster relevant behaviors and eliminates the irrelevant ones. Training such a model requires a dataset of labeled behavior descriptors. The triplet loss is calculated as follows:

$$J(X, y, \beta \mid \Theta) = \frac{1}{N}\sum\nolimits_{(u,v,w)\in triplets(X,y)} -log\left(\sigma(u \cdot v - u \cdot w - \beta)\right), \tag{5.9}$$

where X represents the behavior descriptor matrix, y represents the ground truth label for each descriptor, β is a separation margin for triplet loss, θ represents parameters to be optimized with gradient descent (in our case it represents neural network weights), and N represents the batch size. It must be noted that we assumed mini batches training with stochastic gradient descent.

We sample N triplets (u, v, w) based on the dataset of labeled multiple sensor data streams (X, y). This can be guaranteed at encoder $f(\ )$level – for example, by using *tanh* nonlinearity. Because the function *tanh* can have values only in the range [-1, 1], the values of u, v, and w are restricted to this interval. To obtain the behavior descriptor, we use encoder neural network $f(\ )$:

$$h = f\left(X\big(t_{i,j}\big)\right), \tag{5.10}$$

where $t_{i,j}$ corresponds to the number of i_{th} MoLD-S associated with a triplet component number j, $i = \overline{1,N}$, $j = \overline{1,3}$, $y\big(t_{i,1}\big) = y\big(t_{i,2}\big)$, and $y\big(t_{i,1}\big) \neq y\big(t_{i,3}\big)$. The concrete algorithms developed and used for the implementation of the merging MoLD-Ss sub-function are presented in Section 5.4.1.

5.3.2. Fundamentals of the implementation of the task-relevant data analytics tools recommendation module

Currently, recommendation systems are highly popular and are effective for information filtering [61]. They are widely deployed to address the challenge of overwhelming amounts of information [62]. For this reason, we decided to build a task-relevant data analytics tools recommendation module. The intention was to filter the overwhelming amount of information (by the data analytics tools) for designers based on their design tasks.

To retrieve the DST with the minimum distance to the DT given by the designer, the first step is to calculate this minimum distance between the textual formulations of DSTs and the DT. In the literature, various techniques and functions have been used for this purpose, such as (i) Euclidean distance [63], (ii) pattern-based distance [64], and (iii) edit distance [65]. Euclidean distance is considered as the basis of a number of measures of similarity and dissimilarity [66]. The Euclidean distance between vectors X and Y (between a query and a text, for example) is defined as follows [67]:

$$d(X,Y) = \sqrt{\sum_{t=1}^{n}(X_t - Y_t)^2}, \tag{5.11}$$

where the set of terms is $T = \{t_1, t_2, \dots, t_n\}$, and X_t and Y_t are terms' weights.

In other words, it is the square root of the sum of squared differences between corresponding elements of two vectors. It is important to mention that Euclidean distance treats the values of X and Y as they are. This means that no adjustment can be made for differences in scale. Consequently, this distance calculation is only appropriate for data measured on the same scale [68]. On the other hand, pattern-based distance is defined as the Euclidean distance between sequences based on their moving averages. It can handle sequences with different baselines, scales, and time offsets [69]. For two sequences, s_i and s_j, of length L, if their structural numerical vectors are S_i and S_j, then the pattern-based distance between s_i and s_j is

$$PDist_L(s_i, s_j) = \max_{a,b \in [1,L]} \left|(S_{ia} - S_{ib}) - (S_{ja} - S_{jb})\right|. \tag{5.12}$$

Consequently, a smaller distance indicates a greater similarity. The process of finding similar patterns requires computing for every pair of sequences and positions. Thus, the computation cost is high for a large number of long sequences [70]. A widely-used notion of string similarity is edit distance (also called Levenshtein distance [71]). It is one of the most widely used metrics to tolerate typographical errors [72]. It calculates the minimum number of insertions, deletions, and substitutions required to transform one string into another one [73].

For building the DSTs' vectors, as prescribed in Table 5.2, the textual descriptions of DSTs must be converted into vectors of words. They are then compared with constant character vectors representing the relationships between DSTx and their characteristics. Two functions are needed for this matter: (i) *strsplit* to split strings (DSTx) and (ii) *strcmp* to compare them. The *strsplit* function splits character strings. The function returns the split results in a list, where each component of the list is the split results of DST [74]. The DST is a vector of characters, and "split" is a character vector containing regular expressions used for the split.

The function *strcmp* performs binary safe string comparisons (as defined in the *PHP: Hypertext Preprocessor*) [75]. In considering two strings, the function returns 1 (true) if they are identical and 0 (false) otherwise. The texts are considered identical if the size and the content of both are the same. The input arguments can be any combination of string arrays, character vectors, or cell arrays of character vectors [76]. Concerning the building of DATs' vectors, we suppose having a characterization of the tools based on the data source, data categories, and expected data outputs. Thus, a simple transformation is needed to build tools' vectors, including their DS, DC, and DO characteristics, as fulfilment of the considered three criteria. This simple transformation is presented in Section 5.4.2.

After obtaining both the DATs' and DSTs' vectors, the distances between them must be calculated to determine the corresponding tool for a specific task. To this end, we need to determine the patterns within the vectors. In practice, this means finding strings within other strings using *strfind* and checking whether the array containing the strings is empty by using *isempty*. This function returns 1 if the array is empty and 0 otherwise [77].

To sort DATs related to a specific DST, we adopted the so-called "bubble sort algorithm" (sometimes referred to in the literature as the sinking sort algorithm). It is a sorting algorithm that steps into the list to be sorted repeatedly, compares every pair of adjacent items, and changes their order if the order is wrong. The algorithm repeats its pass through the list until no swaps are needed [78]. The results is that the list is sorted. The bubble sort algorithm is simple and easy to implement [79]. The sorting is done using the weights of DATs. The last operation of the recommendation function proposes a (final) finite set of the best-matching DATs. We decided to do it based on the maximum of the summed weights for the selected tools. For this reason, a function *max* is used to determine the tools with the maximum weight sum (those to be presented as final output for the designer). The implementations of all algorithms needed for the realization of the task-relevant data analytics tools recommendation module are presented in Section 5.4.2.

5.3.3. Fundamentals of the implementation of the smart user identification module

Smart user identification relies on face identification and recognition; as such it is not a novel function in terms of its implementation. However, we deemed it to be novel in the case of development of data analytics tools (or toolboxes). In the literature, many efforts and solutions have been proposed for the realization of this function in terms of algorithms and data constructs. Our objective was to study existing approaches and arrive at a suitable means of implementing F_{SI}.

Biometric recognition methods are widespread and of great interest in many fields, such as security, protection, financial transaction verification, airports, and office buildings [80]. In the last few decades, face recognition became one of the most important applications of biometric recognition systems [81]. Face recognition systems fall into one of two classifications: verification or identification [82]. Face verification is a one-to-one matching that compares the face from the image with a template face image of the person whose identity is being claimed. Face identification, by contrast, is a one-to-many matching that compares a query face image against all image templates in a face database [83]. After detecting a face from the acquired image, a face recognition system preprocesses the image to convert the face image to grayscale and resizes it to reduce its dimensions, if required. Features extraction is then performed, regardless of the lighting, expression, illumination, aging, rotation, image scale, and pose [84]. Face recognition is a complex process because its computational model is difficult, and faces are complex, multidimensional, and meaningful visual stimuli [85].

Among the many face detection methods presented in the literature, the Viola–Jones algorithm is considered the most successful in terms of accuracy and speed in visual object detection [86]. It is also the most widely used algorithm in face detection [87]. The Viola–Jones algorithm is intended for real-time face detection from a general image. It has several advantages, such as a high detection rate, integral image representation that allows quick feature computation, coherent feature selection, high rejection of non-facial images, and invariance to small changes in scale and location [88]. Its real-time performance is ensured by using Haar-type features computed using (i) integral images, (ii) classifier learning with the Adaptive Boost (AdaBoost) algorithm, and (iii) face detection using attentional cascade structure [89]. Integral image is a preprocessing step. It converts the input face image into an integral image by making each pixel of the image equal to the entire summation of all pixels above and to the left of the given pixel [90]. The advantage of using an integral image is that it increases the speed of features extraction [91]. Calculating an integral image is represented in the equation below:

$$I(x,y) = \sum_{x' \leq x, y' \leq y} O(x', y'), \tag{5.13}$$

where I represents the integral image, O represents the original one, and (x, y) and (x', y') are pixel coordinates belonging to them.

Past experience has shown that it is efficient to complete the summation of pixels in any rectangular area using the integral image. The summation in a rectangular area $Z = [A_1, A_2, A_3, A_4]$ can be calculated as follows:

$$I(x,y) = \sum_{(x,y) \in Z} O(A_4) + O(A_1) + O(A_2) + O(A_3), \tag{5.14}$$

where $[A_1, A_2, A_3, A_4]$ are the segments of the rectangular area, assumed to have positive geometric coordinate values, as shown in Figure 5.1.

The features are calculated considering that the summation of the pixels can be computed in the constituent rectangles in constant time. It has been observed that a detector with a basic resolution of 24 × 24 pixels can produce positive results [92]. Even if a rectangular feature is calculated in constant time, it is not sensible to calculate every 24 × 24 pixels over 160 000 features in a real-time operation. To work efficiently with features, only the best features have to be selected. This can be achieved by using the AdaBoost algorithm, which picks up the best features and uses them to train strong classifiers [93]. In addition, the AdaBoost algorithm is also useful at selecting a set of classifiers from a family of weak classifiers $\{C_\omega\}$, where ω is the compound parameter. For each image O, the classifier C_ω elaborates a hypothesis $\delta_\omega(o) \in \{-1, +1\}$ on the

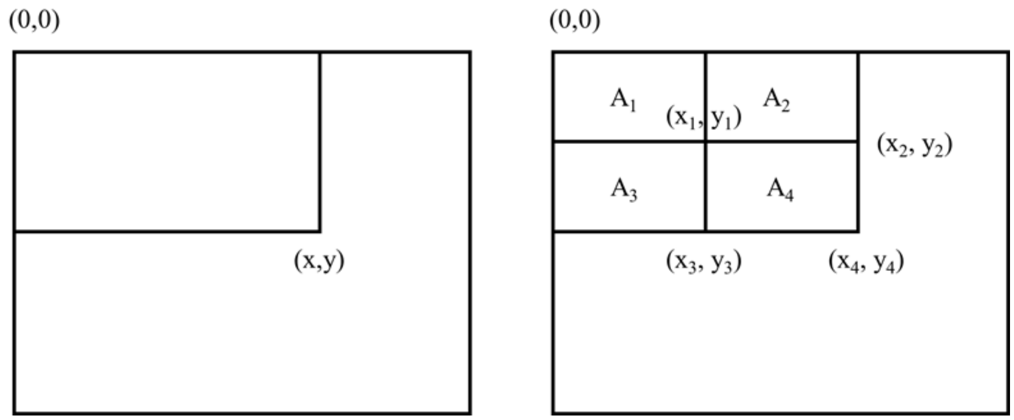


Figure 5.1. Interpretation of A_1, A_2, A_3 and A_4

membership of O in one of two classes labeled by −1 and +1. In the case of $\delta_\omega(o) = 1$, the cost of such a decision is $\gamma_\omega(O) = \alpha_\omega$; otherwise $\gamma_\omega(O) = \beta_\omega$. The cost of the decision is a real number, and it can be negative.

As mentioned above, the AdaBoost algorithm selects the best classifier ω_{best}, which has the minimum of the average classification error for the training sequence (O_1, w_1, y_1), ..., (O_L, w_L, y_L), by calling a procedure *getBestWeakClassifier* (for simplification: *getBWC*), as shown below:

$$\omega_{best} \leftarrow \arg min\, \epsilon(\omega) \triangleq \arg\min \sum_{i=1}^{L} \mathrm{w}_i \,|\delta_\omega(O_i) - y_i|\,. \tag{5.15}$$

If the algorithm selects the classifiers $C_{\omega 1}$, …, $C_{\omega M}$, the strong classifier $C_{\omega M}$ elaborates a hypothesis $\Delta(O)$ for O by summing the costs of individual decisions and comparing the results with the value 0:

$$\Delta_M \triangleq \begin{cases} +1, & if \displaystyle\sum_{m=1}^{M} \gamma_{\omega\mathrm{m}}(O) > 0 \\ -1, & otherwise \end{cases} \quad . \tag{5.16}$$

AdaBoost calls *getGoodDecisionCosts* (for simplification: *getGDC*) to determine the costs of weak decisions. This function returns the cost α for the positive hypothesis and the cost β for the negative hypothesis (α and β are classes). In Section 5.4, the detailed AdaBoost algorithm is presented. To achieve a higher detection rate and a lower misclassified image detection rate, another strong classifier should be used to correctly reclassify the earlier misclassified images. According to the terminology of the Matlab cascade object detector, this creates the attentional cascade. At the first layer of the attentional cascade, a strong classifier with few features is used, which will filter and reject most negative windows [94].

As discussed by Alionte, E. and Lazar, C., “The file is created with the help of the trainCascadeObjectDetector function. The attentional cascade training is done using a set of positive samples (windows with faces) and a set of negative images. For obtaining a more accurate detector, the number of cascade layers and the function parameters were tuned. Finally, for different tuning parameters the performances of the face detector were analyzed" [89]. Becoming more and more complex, the used cascade of classifiers will achieve a better detection rate. At every layer of the cascade, the correctly classified negative images will be eliminated, and the new strong classifier will have a more difficult task than the previous step classifier [95]. The steps of operation of the cascade classifier are the following: (i) the image is split into multiple windows; (ii) each window is an input in the attentional cascade; (iii) at every layer, the window is checked for whether it contains a face (according to the strong classifier); (iv) if the result is negative, the window is rejected, and the previous steps are repeated for the following windows; (v) if the result is positive (possible face), it is taken to the next layer of the cascade; and, in the end, (vi) a window is considered to have a face if it passes through all layers of the attentional cascade [96]. The details of the training algorithm for building a cascade detector can be found in [97] and [98].

Many techniques for concrete face recognition are presented in the literature, such as (i) principal component analysis [99], (ii) independent component analysis [100], (iii) support vector machines [101], (iv) linear discriminant analysis [102], (v) hidden Markov models [103], (vi) DNNs [104], and (vii) local binary patterns (LBPs) [105]. In a comparison of these methods, the LBP method has proven to be the most effective for face recognition [106]. It improves the precision of face recognition, especially when combined with edge histogram descriptor (EHD) [107]. EHD captures the spatial distribution of the edges of images, providing a set of standard tools to describe an image [108]. It was selected for detecting features for face representation [109]. Those features are (i) translation and scale invariant, (ii) less sensitive to noise and illumination, and (iii) have low dimension [110]. EHD reduces storage requirements and eases the subsequent computation [111]. Face recognition is no longer a high-dimensionality problem when faces are represented using EHD features [112].

As mentioned above, the LBP is the most popular feature descriptor for face recognition and textual classification [113]. It has a tolerance for monotonic illumination changes and is computationally simple. It summarizes local structures of images by comparing every pixel with its neighbor [114]. In LBP, a facial image is divided into logical regions. A texture descriptor is extracted from these regions, which are concatenated to form a global description of the face [115]. LBP valuates each pixel into a binary digit (either 0 or 1). Given a central pixel value P_c in an image, its neighboring pixels (P_0, P_1,…, P_{i-1}) are selected using a radial filter [116]. The response at P_c is calculated using the following equation:

$$LBP = \sum_{i=0}^{i-1} s(P_i - P_c)2^i \quad , \quad where: \; s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}. \qquad (5.17)$$

The coordinates for every sampling iteration are calculated using a circular coordinate system with center P_c. The coordinates are given by $(Rcos(2\pi i/P), Rsin(2\pi i/P))$, where P is the total number of involved neighboring pixels, and R is the radius around center pixel. The histogram is afterwards built to represent features by using the previously calculated LBP as follows:

$$H(k) = \sum_{i=0}^{i} \sum_{j=0}^{j} f\big(LBP_{P,R}(i,j), k\big), where: k \in [0, K]\ ;\ F(x,y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}. \quad (5.18)$$

In Equation 5.18, K represents the maximum LBP value. The histogram also contains information about the distribution of edges, spots, and flat areas over the whole image.

In our work, we integrated a color structure descriptor (CSD) with the texture descriptor. The CSD is a color histogram in the quantized Hue–Max–Min–Diff (HMMD) color space. In comparative studies of color descriptors, the CSD achieved the best performance [117]. The HMMD color model is well suited for image retrieval. It has five parameters. Hue is expressed from 0˚ to 360˚ in the Hue region. When the angle changes, Hue becomes red (0˚ = 360˚), green (120˚), or blue (240˚). Max describes the quantity of black and gives shades of color. Min is the quantity of white and gives tints of colors. Diff is the quantity of gray and gives tone. A function, *Sum*, is used to calculate the brightness of the colors. Hue, Max, and Min or Hue, Diff, and *Sum* are sufficient to analyze the distribution of color space [118].

Edge histograms for an image are constructed by combining sub-images into three types of histograms: local (L-EHD), semi-global (SG-EHD), and global (G-EHD). Generally, L-EHD represents the spatial distribution of different types of edges, while S-EHD and G-EHD are obtained by combining histograms of local structures [119]. EHD involves dividing an image into 4 × 4 sub-images. Edges are grouped into four categories: (i) horizontal, (ii) vertical, (iii) diagonal (including both 45˚ and 135˚ diagonals), and (iv) anti-diagonal (also called nondirectional) [120]. To extract these four edges, each sub-image is divided into nonoverlapping square blocks of 2 × 2 pixels. In each of the blocks, four edge-oriented filters are applied to compute the edge strength. The edge strength of every image block is calculated to determine the type of edge. If the maximum value of edge magnitude is greater than a threshold value, it is assigned an edge type.

After the features are extracted, the next step is measuring the similarity between the query image and the images stored in the database of the SDATB. This step can be done using the image similarity function S_I, which represents the sum of score fusion for each feature. The reader is reminded that the main objective of face recognition is to match one face image to another and to assign a matching score to determine the possibility that two compared face images are of the same person. It is often the case that we have multiple matching scores due to, for example, multiple medias for a face, multiple matching methods, or multiple features. Accordingly, we need to obtain a single score for the considered matching pair using score fusion.

Mathematically, score fusion can be computed with the following equation [121]:

$$S_I\big(i_j, \mathrm{q}\big) = \sum_{f_n \in F} \left[\frac{S_F(i_j, q; f_n)}{svd(S_F(I, q; f_n))}\right], \quad (5.19)$$

where $S_F(i_j, q; f_n)$ is feature similarity between the image i_j and query q with respect to feature f_n, and $S_F(I, q; f_n)$ stands for all of the feature similarity values between i_j and query q with respect to feature f_n. If we have N_I images in the database, then this value forms a vector of feature similarity with the size $N_I \times 1$. Furthermore, $svd(S_F(I, q; f_n))$ is a function to get the singular value decomposition of $S_F(I, q; f_n)$, I is the symbol of the image, i_j is the elements of the image, N_I is the size of the image, F is the symbol of the feature extracted from an image, f_n is the element of the feature, and N_F is the size of the feature.

Considering a feature vector of an image i_j with respect to feature f_n is $X_p = (x_1, \ldots, x_D)$ and the feature vector of q with respect to feature f_n is $Y_p = (y_1, \ldots, y_D)$, the similarity feature between image "" and query q with respect to feature f_n is defined as

$$S_F(i_j, q; f_n) = \begin{cases} 1 - \dfrac{\sum_{p=1}^{D}(X_p, Y_p)}{\min\left(\sum_{p=1}^{D} X_p, \sum_{p=1}^{D} Y_p\right)}, if\ f_n is\ LBP \\ \left(\displaystyle\sum_{p=1}^{D}\left(X_p - Y_p\right)^2\right)^{1/2} \qquad , otherwise \end{cases} . \quad (5.20)$$

The source of Equation 5.20 is the article by Kusumaningrum, R. and Arymurthy, A.M. [121]. When histograms are extracted from a set of images, they are compared to each other to determine their similarity [122]. In the implementation Section 5.4.3, the chunks of knowledge presented above are used to design algorithms, including the basic algorithms for image acquisition and database building.

5.4. Implementation of the computational mechanisms of the demonstrative data analytics toolbox

5.4.1. Implementation of the merging middle-of-life data streams recommendation module

To realize the merging MoLD-Ss recommendation module, the algorithms presented in Table 5.1 need to be implemented. To acquire real-time MoLD-Ss to be analyzed and merged, three algorithms, A_1, A_2, and A_3, are needed. This is also the case for some of the algorithms presented in this dissertation. They are either newly developed by software tools such as Matlab, Python, or SPSS, or they are known – that is, described in the literature – such as Algorithm A_6, called the "sliding window algorithm" [123]. In this sense, we detail only the new algorithms that we designed to realize the proposed functions; the remaining ones are explained and used explicitly in Section 5.5.1. To explore data and provide means for a designer to visualize the desired MoLD-Ss (which have been transferred from remote cloud storage or local storage that aggregates and saves IoT information), Algorithm A_4 has been developed to generate an interface for MoLD-Ss visualization.

For this algorithm, we need to define the following: (i) Matrix D of M x 2 x S x T dimension with instances of sensors' time series data. The first dimension M corresponds to sensor number, the second dimension corresponds to either normal behavior (-1) or faulty behavior (-2), and the third dimension corresponds to S = 256 instances of different windows of sensor data, each of which has T = 256 time steps (representing the fourth dimension). (ii) *SensorsNames* is a function providing sensor descriptions for each of the M available sensors. (iii) The *RequestIds* function is responsible for visualizing a chosen set of sensors at the same time. Once the visualization is finished, the function returns an empty set (nothing to visualize). (iv) The *PlotTimeseries* function is a user interface method to display multiple sensor data streams within the same window in a certain time range.

Algorithm 4. Plot sensors data streams as time series for selected data streams.

Inputs: I1 = D
I2 = SensorNames function
I3 = RequestIds function. UI method to request a subset of sensors
I4 = PlotTimesseries
Outputs: PlotTimeseries(A, SensorNames)

$sel \leftarrow RequestIds(SensorNames)$; % obtains sensors selected by user
if $numel(sel) == 0$
  $break$;
end
$A \leftarrow zeros(numel(sel), 256)$; % output matrix to pass into UI method
for $i \leftarrow 1{:}numel(sel)$
    $t18 \leftarrow randi([1,2])$; % randomly select either time series are faulty or not
    $t17 \leftarrow randi([1,S])$; % randomly select one of S time series instances
    $A(i,:) \leftarrow squeeze(D(i,t18,t17,:))$;
end
Return PlotTimeseries(A, SensorNames)

To provide a recommendation based on a multi-stream dataset *D'*, we specify annotations to past anomalies. Descriptions of past anomalies need to be specified. Let us consider a window of aligned multi-modal features $X = \{X(t, k)\}$, where $t = [a, b]$ and $k = [1, M]$. M is the total number of selected sensors. The interval $[a, b]$ represents the time boundaries of the anomalous behavior of the historical data of some device. Since we considered the triplet loss function for the ANN used for clustering a predefined set of classes, we assigned unique labels to the anomalies. Furthermore, we defined a set of incidents for each anomaly to allow the model to have sufficient data during the training and to avoid overfitting. The triplet loss training is capable of fitting a dataset of 8 million unique labels and achieving > 95% classification accuracy [124]. The neural networks architecture that was considered for this purpose can be described as the algorithm responsible for sensor importance weight predictions (A_7) for forward pass (which refers to the calculation process and values of the output layers from the input data). To build the algorithm, we needed to define a real=valued matrix X with B x M x T size, where M represents multimodal features of each window of frames (sliding window), T represents the time frame, and B is the batch size.

Algorithm 7. Estimate sensors importance

Inputs: I1 = Matrix X
I2 = B
Outputs: O1= h, latent representation of behavior described by the current window of features
O2 = a, sensors importance

$t2 \leftarrow conv1(X)$;

$t16 \leftarrow leaky\_relu(t2, 0.2);$
$t15 \leftarrow conv2(t16);$
$t7 \leftarrow leaky\_relu(t15, 0.2);$
$t6 \leftarrow attention(t7);$
$t8 \leftarrow reshape(t6, B, M, 1) .* \ t7;$
$t9 \leftarrow behavior\_conv1(t8);$
$t10 \leftarrow tanh(reshape(sum(t9, 2), B, 1, L));$
Return *struct('h', t10, 'a', t6)*

To train the model, we used a specific triplet loss algorithm, known in the literature as a hinge triplet loss algorithm [125]. This algorithm uses a hinge function to create a fixed margin between the anchor-positive difference and the anchor-negative difference. The following inputs had to be defined: (i) H, a real-valued matrix of B x 3 x L dimension, where B is the batch size, 3 represents two triplets of same label behavior representations and one outlier, and L is a latent representation dimension. H must be constrained within the boundaries [-1, 1]; otherwise, either a *tanh*(.) activation function can be applied, or rescaling of the vector values can be considered. (ii) *B'*, a separation margin to control how much nonrelevant behavior should be embedded in the latent space according to cosine similarity distance. This Algorithm (A_9) is presented below:

Algorithm 9. The estimate behavior descriptor based on the merged MoLD-Ss representation

---

Inputs: I1 = H
I2 = B'
Outputs: O1= J, loss value that is to be minimized with a gradient descent algorithm
O2= Acc, separation accuracy of triplets within specified margin

---

$t1 \leftarrow sum(H(:,1,:) .* \ H(:,2,:), 3) - sum(H(:,1,:) .* \ H(:,3,:), 3) - B';$
$t2 \leftarrow sigmoid(t1);$
$t3 \leftarrow -log(t2);$ % we maximize likelihood of t2 probability to be equal to 1
$t4 \leftarrow mean(t2 > 0.5);$
Return struct('J', mean(t3, 1), 'Acc', t4)

The triplet loss presented in Algorithm A_9 has been optimized using stochastic gradient descent (SGD) [126]. This algorithm optimizes the triplet loss by changing the parameters of the neural network. Accordingly, the following inputs were defined: (i) W is a list of weight matrices created by the neural network model, W (i) is a particular multidimensional weight matrix determined by the neural network model, (ii) *D* is a dataset of samples to be used for mini batches of data generation, (iii) *lr* stands for the learning rate parameters that control step size along the direction opposite the loss function gradient, (iv) epochs are the numbers of iterations to perform across the dataset, and (v) *epoch_size* is the number of gradient descent steps per epoch.

Algorithm. SGD

Inputs: I1 = W
I2 = D

I3 = lr
I4 = Epochs
I5 = Epoch_size

Outputs: O1= logs, learning curves for train/test splits of the data, as well as separation accuracy matrix evaluation
O2= best_weights, trained neural network parameters

```
for w = W
    w ← random weights initialization();
end
logs ← {};
best_eval ← +Inf;
best_weights ← W;
for epoch = 1: epochs
    for split_type = ['train', 'test']
        for step = 1: epoch_size
            batch ← sample B triplets of windows of features
            from D.(split_type), i.e. X;
            H ← BehaviourEncoder(batch.X);
            J, Acc ← TripletLoss(H);
            if split_type == 'train'
               for w = W
                   w ← w − lr ∗ Jw' ; % derivative of Triplet Loss(H|W)
                   regarding parameter matrix w
               end
            end
            logs.(split_type)(epoch).accuracy ← Acc
            logs.(split_type)(epoch).triplet_loss ← J
        end
    end
    if mean(logs.test.(epoch).triplet_loss) < best_eval
      best_eval ← mean(logs.test.(epoch).triplet_loss)
      best_weights ← W
    end
  Return struct('logs', logs, 'best_weights', best_weights)
```

During the stochastic gradient descent procedure, we sampled a batch of triplets to perform the optimization step. The following inputs were needed: (i) D is M x 2 x S x T matrix with instances of the time series data generated by the sensors, where the first dimension corresponds to the number of sensors, the second corresponds to either normal (1) or faulty (2) behaviors, and the third dimension corresponds to S = 256 instances of the different windows of sensors data, each of them having T = 256 time steps, (ii) B is the batch size, (iii) C is a causality matrix of size N x M, which contains N behavior patterns obtained from M sensors, that may affect it by being faulty.

Algorithm. Sample batch of triplets

Inputs: I1 = D

I2 = B
I3 = C

Outputs: O1= X, is Bx3xMxT matrix with B triplets of windows, with T time steps, and having M different sensor streams

```
t4 ← zeros(B, 3, M, T); % output matrix
t17 ← zeros(B, 3); % triplet of pattern/behavior ids to sample
for i ← 1:B
    t8 ← randi(N + 1); % anchor pattern id
    t12 ← 1 : (N + 1);
    t13 ← (t12 ~= t8); % not select patterns mask
    t14 ← t12(t13); % not selected pattern ids
    t15 ← randi(numel(t14));
    t16 ← t14(t15); % outlier pattern id
    t17(i, 1:2) ← t8; % select t8 as pattern id for first and second items in the
    triplet
    t17(i, 3) ← t16; % select t16 as an outlier pattern if in the triplet, namely
    third item
end
for i ← 1 : B
    for j ← 1 : 3
        for k ← 1 : M
        t4(i, j, k, :) ← squeeze(o_19(k, 1, randi([1, S]), :)); % initialize each
        sensor with normal behavior time series first
        end
    if t17(i, j) ~= N + 1 % verify that current pattern id has entry in causality
      matrix
       t18 ← t17(i, j);
       t10 ← find(C(t18, :) > 0); % find nonzero entries in causality matrix
       for pattern t18, which correspond to sensor ids
       for k ← t10
           t4(i, j, k, :) ← squeeze(o_19(k, 2, randi[1, S]), :)); % sample faulty
           behavior for each pattern shaping sensor
       end
    end
end
 Return struct('X', t4)
```

After introducing the behavior descriptors of multiple sensor data streams, we developed Algorithm A_{30} to select potential candidates for an anomaly. For this purpose, three inputs have to be defined: (i) h, a matrix of size $N \times L_3$ of behavior descriptors to analyze; (ii) q, a matrix of size $M \times L_3$ of behavior descriptors in the database of past anomalies; and (ii) *tau*, the upper bound of the confidence interval. The detailed Algorithm A_{10} is presented below:

Algorithm 10. Estimate probability of the anomaly

Inputs: I1 = h

I2 = q
I3 = tau

Outputs: O1 = p, N dimensional vector specifying the probability that N exhibits anomalous behavior

```
for i = 1:N
    p(i) ← 0.0;
    for j = 1:M
        cur_p = σ(h_i · q_j');
        if cur_p > p(i)
            p(i) ← cur_p;
        end
        if p(i) > tau
            break;
        end
    end
end
Return struct('p', p)
```

Given the vector p, we select as candidates only those entries for which $p_i > \tau$, where τ is the upper bound of confidence interval for normal behavior. Algorithm A_{30} filters out normal cases based on the large number of descriptors, which are generated by the sliding window approach working on time series data. Algorithm A_{10} was also intended to detect anomalies. Another algorithm was developed for similarity-based searching, which is based on similarity estimation. This was done because, in addition to detecting an anomaly, we must also retrieval a ranked list of relevant anomalies for the computational processing. To develop Algorithm A_{11}, the following inputs were defined: (i) h, a matrix of size $N \times L_3$ of behavior descriptors to find similar past cases; (ii) q, a matrix of size $M \times L_3$ of behavior descriptors in the database of past anomalies; and (iii) tau, the distance threshold for descriptor retrieval. The Algorithm A_{11} is presented below:

Algorithm 11. Perform search for similar descriptors in database

Inputs: I1 = h
I2 = q
I3 = tau

Outputs: O1 = index, identifiers of relevant past anomalies descriptors
O2 = index, N dimensional vector specifying an offset of descriptors retrieved for a particular anomaly candidate specified by the array index (Note that Matlab handles every variable as an array that can hold numbers. In order to access selected elements of an array, indexing is used).
O3 = amount, N dimensional vector specifying the number of retrieved the descriptors per anomaly candidate

```
index ← []
offset ← []
amount ← []
```

```
for i ← 1:N
    offset(i) ← numel(index) + 1;
    amount(i) = 0;
    for j ← 1:M
        d ← σ(h_i · q_j');
        if d > tau
            index(offset(i) + amount(i)) ← j;
            amount(i) ← amount(i) + 1;
        end
    end
end
Return struct('index', index, 'offset', offset, 'amount', amount);
```

After determining the possible anomaly candidates, we used Algorithm A_{12} to calculate the distances between these anomalies. This algorithm requires the following inputs: (i) h, a matrix of size $N \times L_3$ of behavior descriptors for anomaly candidates; (ii) q, a matrix of size $M \times L_3$ of behavior descriptors in the database of past anomalies; (iii) index, identifiers of past anomalies; (iv) the offset of the first entry for each of the N anomalies; and (v) the number of relevant past cases discovered for each of the N anomalies. The meta-code of Algorithm A_{12} is shown below:

Algorithm 12. Calculate distance between anomalies

Inputs:	I1 = h I2 = q I3 = index I4 = offset I5 = amount
Outputs:	O1 = d, distance between each of N anomalies and past case relevant to them

```
d ← []
for i ← 1:N
    for j ← 0:amount(i) − 1
        k ← index(offset(i) + j);
        cur_d ← σ(h_i · q_k');
        d(offset(i) + j) ← cur_d;
    end
end
Return struct('d', d)
```

Given the distances of past cases, they can be sorted to generate a ranked list of anomalies. This is achieved with Algorithm A_{13}, presented below. This algorithm needs four inputs: (i) d, distances between anomalies (expressing the degree of similarity between anomalies and the past cases relevant to them); (ii) index, identifiers of past anomalies; (iii) offset of the first entry for each of the N anomalies; and (iv) the total number of relevant past cases found for each of the N anomalies. It represents the similarity between anomalies and past cases relevant to them.

Algorithm 13. Rank anomalies

Inputs: I1 = d
I2 = index
I3 = offset
I4 = amount

Outputs: O1 = r, ranked identifiers of past anomalies
O2 = r_index, list of the identifiers of anomalies in the distance vector d

for $i \leftarrow 1{:}N$
  $a \leftarrow d(offset(i){:}offset(i) + amount(i) - 1)$;
  $[b, i] \leftarrow sort(a)$;
  $c \leftarrow index(offset(i){:}offset(i) + amount(i) - 1)$;
  $r(offset(i){:}offset(i) + amount(i) - 1) \leftarrow c(i)$;
  $r\_index(offset(i){:}offset() + amount(i) - 1) \leftarrow i$;
end
Return struct('r', r, 'r_index', r_index)

To generate a recommendation, we need to obtain the top K anomalies per descriptor using a ranked list of their identifiers. This can be done with Algorithm $A_{14,}$ presented below. The inputs for this algorithm are as follows: (i) r, a ranked list of the identifiers of relevant anomalies; (ii) index, identifiers of anomalies in the distance vector d; (iii) offset of the first entry of each of the N anomalies; (iv) C, an $M \times L_4$ causality matrix of past anomalies related to L_4 sensors; and (v) K, the total number of the (most) relevant anomalies to be found for each of the N candidate anomalies.

Algorithm 14. Retrieve relevant anomalies based on their ranking and the corresponding sensors

Inputs: I1 = r_
I2 = r_index
I3 = offset
I4 = amount
I5 = C
I6 = K

Outputs: O1 = sensors, sensor identifiers for each past anomaly
O2 = sensors_offset, offset of each sensor influenced by anomalies (representing what anomalies to remove or to keep)
O3 = sensors_amount, number of sensors influenced by anomalies
O4 = anomaly, anomaly identifiers with up to K entries per each of the N anomaly candidates
O5 = anomaly_new_index, anomaly identifiers within distance d
O6 = anomaly_offset, offset of each anomaly group
O7 = anomaly_amount, size of each anomaly group

for $i \leftarrow 1{:}N$
  $anomaly\_offset(i) \leftarrow numel(anomaly)$;
  $anomaly\_amount(i) \leftarrow 0$;
  $t\_1 \leftarrow anomaly\_offset(i)$;
  for $j \leftarrow 0 : min(amount(i), K) - 1$

```
            k ← r(offset(i) + j);
            anomaly_amount(i) ← anomaly_amount(i) + 1;
            t_3 ← t_1 + j;
            anomaly(t_3) ← k;
            anomaly_new_index(t_3) ← r_index(offset(i) + j);
            sensors_offset(t_3) ← numel(sensors);
            sensors_amount(t_3) ← 0;
            for l ← 1:L_4
                if C(k,l) = 1
                    t_4 ← sensors_amount(t_3);
                    sensors(t_4) ← l;
                    sensors_amount(t_3) ← sensors_amount(t_3) + 1;
                end
            end
    end
    Returns struct('sensors', sensors, 'sensors_offset', sensors_offset,
            'sensors_amount', sensors_amount, 'anomaly', anomaly,
            'anomaly_offset', anomaly_offset, 'anomaly_amount',
            anomaly_amount);
```

The database contains “if … then” type rules, which are used in mapping between anomalies and possible recommendations. Algorithm A_{15} is used to determine the best match and what to extract. This algorithm requires the following inputs: (i) *d*, distance between anomalies; (ii) sensors, sensor identifiers for each past anomaly; (iii) sensor_offset, offset of each past anomaly sensors list; (iv) sensors_amount, number of each past anomaly sensors, (v) anomaly, anomaly identifiers with up to K entries for each of N anomaly candidates; (vi) anomaly_new_index, anomaly identifiers within retrieved distances of vector d; (vii) anomaly_offset, offset of each anomaly group; (viii) anomaly_amount, size of each anomaly group; and (ix) sensors_importance, matrix of size $N \times L_4$ of importance weights extracted from attention layer for each anomaly candidate.

Algorithm 15. Identification of possible actions (recommendation)

Inputs: I1 = d
I2 = sensors
I3 = sensor_offset
I4 = sensors_amount
I5 = anomaly
I6 = anomaly_new_index
I7 = anomaly_offset
I8 = anomaly_amount
I9 = sensors_importance

Outputs: O1 = faulty_sensors, identifiers of the sensors that most likely cause anomaly
O2 = anomaly_action: identification of possible actions (recommendation) matching the most relevant past anomalies with

the smallest distance to the detected anomaly candidate

```
best_match ← −1;
best_distance ← +Inf;
for i ← 1 : N
    if anomaly_amount(i) = 0
      continue;
    end
     k ← anomaly_new_index(anomaly_offset(i));
     if d(k) < best_distance
        best_match ← i;
        best_distance ← d(k);
     end
end
if best_match = −1
   Return struct('faulty_sensors', {}, 'anomaly_action', {});
if sensors_amount(anomaly_offset(best_match)) > 0
    t_1 ← sensors_offset(anomaly_offset(best_match));
    t_2 ← sensors_amoun(anomaly_offset(best_match));
    t_3 ← sensors(t_1: t_1 + t_2 – 1);
    t_4 ← sensors_importance(t_3);
    [t5, t6] ← sort(t4);
     faulty_sensors ← t3(flip(t6));
end
 t_7 ← anomaly_offset(best_match);
 t_8 ← anomaly_amount(best_match);
 anomaly_action ← anomaly(t7: t7 + t8);
 Return ('faulty_sensors', faulty_sensors, 'anomaly_action',
                 anomaly_action);
```

The other algorithms not presented in this section are used during the functional validation of the merging MoLD-Ss recommendation module, discussed in Section 5.5.1.

In addition to the developed algorithms, the computational workflow (CWF) is also an important characteristic of this recommendation module for merging MoLD-Ss. Ordering all computational steps, the CWF of this module is shown in Figure 5.2. After the sensors are located and the data streams for analysis and merging are selected, the data contents of the MoLD-Ss are transferred to the SDATB, as step that completes the analysis and the merging. In the next step, the data contents are visualized and presented in various plotted forms to the designer. The designer is given the opportunity to visualize the stream plots more than once. Towards the merging, the sliding window algorithm is used to iterate over the MoLD-Ss. The algorithm selects a consequent time frame of data and normalizes the data along the time axis. After this, the single-stream encoder part of the used neural network is applied, and single-sensor latent representation is generated in the attention layer of the neural network.

In the next step of the data processing, the single-sensor representation is rescaled according to the importance weights. These rescaled representations are concatenated into a two-dimensional matrix, and the behavior encoder part of the neural network is applied. Furthermore, the toolbox queries the database to find the past anomalies that are closest to the current descriptor. If the distance to past anomalies stored in the database is small, then a confidence interval including the current time window and its descriptor is selected as an anomaly candidate. Otherwise it is skipped. When the algorithm finds no additional windows to analyze, it starts a similarity search. In this context, the descriptors are compared to those stored in the SDATB database. The distances between the anomaly pairs are estimated, and the matches are sorted according to the distances.

After this step, a ranked list of anomaly candidates is retrieved from the database. In combination with this, the sensors relevant to past anomalies are obtained based on the causality matrix. The anomaly candidate that has the shortest distance to its first relevant past anomaly is selected. In terms of the best candidate, this module of the SDATB

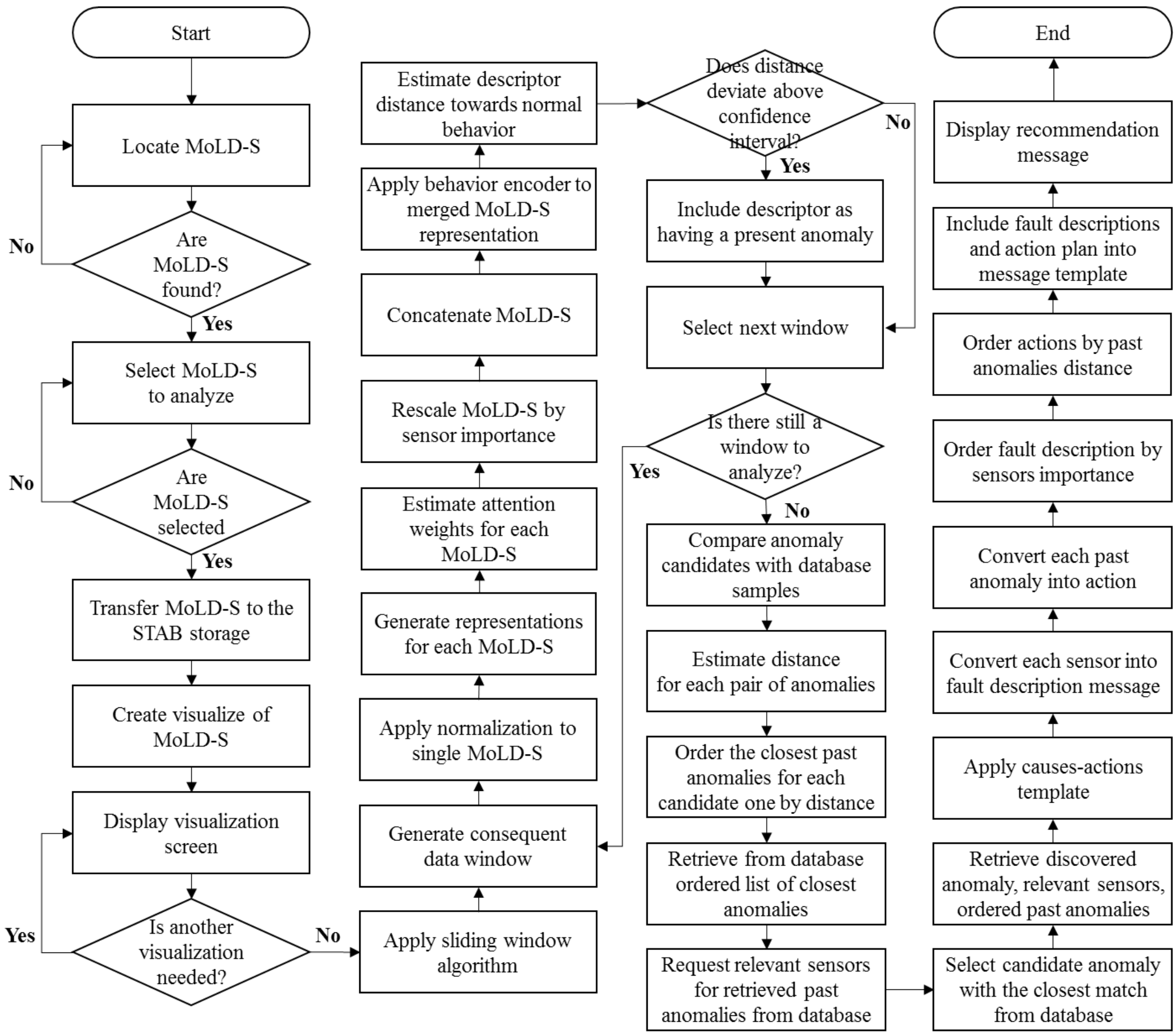


Figure 5.2. The computational workflow of the merging of middle-of-life data streams recommendation module

presents a ranked list of past anomalies, as well as the sensors related to the past anomalies ordered according to the importance weights of the sensors. As a next step, the module selects a template for generating a recommendation message about the faulty sensors and possible improvement patterns. Then, the fault descriptions for each selected sensor and the improvement (or maintenance) actions for each anomaly are retrieved. These are subsequently arranged according to the importance of the sensors and the anomaly distance values and are used to generate the final recommendation message, which includes both the identified faults and the action plan. As the final computational action, this message is displayed to the designer.

5.4.2. Implementation of the task-relevant data analytics tools recommendation module

The CWF of the task-relevant data analytics tools recommendation module starts by recognizing the DT specified by the designer (referred to as DTX in the algorithm development and computational implementation sections). To enable this, the minimum distances between the DT of the designer and the available DTs in the system need to be defined. The *EditDstance* function is used to calculate these distances, which interprets the matching as a standard dynamic programming problem. Given two strings, s_1 and s_2, (e.g. words and sentences), the *EditDistance* function interprets the minimum distance between s_1 and s_2 as the minimum number of operations required to convert string s_1 into string s_2. The computational algorithm of this function is defined as follows:

```
function [V, v] ← EditDistance(string1, string2)
    m ← length(string1);
    n ← length(string2);
    v ← zeros(m + 1, n + 1);
    for i ← 1:1:m
        v(i + 1,1) ← i;
    end
    for j ← 1:1:n
        v(1, j + 1) ← j;
    end
    for i ← 1:m
        for j ← 1:n
            if (string1(i) == string2(j))
                v(i + 1, j + 1) ← v(i, j);
            else
                v(i + 1, j + 1) ← 1 + min(min(v(i + 1, j), v(i, j + 1)), v(i, j));
            end
        end
    end
    V ← v(m + 1, n + 1);
end
```

In our case, s_1 is the DT specified by the designer. As instantiation of s_2, all DTs stored in the database of the SDATB are considered. This latter is done by defining a character array "char," which is in fact a sequence of vectors (textual, numerical). The objective is not only to do the calculation with the EditDistance function, but also to find the minimum distance between DT specified by the designer and "char" of the DTs known to the system (general input number 1: I_1) and DST (I_2). The output is the DST closest to the specified DT. For the execution of the entire procedure, Algorithm A_{21} was implemented.

Algorithm 21. Retrieve DST with minimum distance to DT specified by the designer

Inputs: $I1 = \{DT_1, DT_2, ..., DT_n\}$
$I2 = \{DST_1, DST_2, ..., DST_n\}$
I3 = DTX specified by the designer
I9 = Function "EditDistance"

Outputs: DST_x closest to I_3

```
DTX ← 'I3';
str ← char(I2(1)); % str = char (I2(idx1))
s ← 1000;
aux ← str;
for idx1 ← 1:length(I2)
    str ← char (I2(idx1));
    if (EditDistance (DTX, str) < s)
        s ← EditDistance (DTX, str);
        aux ← str;
        IND ← idx 1;
    end
end
if (IND < numerical value 1)
    DTX ← DT1; % retrieving the index of the minimum if stated
elseif (IND < numerical value 2)
    DTX ← DT2; (continue until DTn-1)
else
    DTX ← DTn;
end
Return DSTX
```

In order to determine the characteristics, patterns have to be recognized. For example, if the retrieved expression of DST_x starts with the word "analyzing" then DST_x can have one of the two alternative characterizations, $DC_x = DC_1$ and $O_x = [O_1, O_2]$, else $DC_x = DC_2$ and $O_x = [O_6, O_8, O_{10}]$. For the realization of the sub-function $F_{SA1,2}$, an implicit transformation matrix is used, which enables a slightly more complicated pattern recognition based on DST_x. The computational procedure converts the textual description of DSTs to a vector of words, and then compares it with the constant character vectors representing the relationships between DST_x and DS_x. Two functions are needed for this computation, namely, (i) strsplit, to split strings (DST_x), and (ii) strcmp, to compare them. The output of $F_{SA1,2}$ is a DST vector including DS, DC and DO. Algorithm A_{22} was implemented to do this computation. The above-mentioned example is used to simplify the writing of the algorithm.

Algorithm 22. Build DSTs vectors

Inputs: I4 = {DS_1, DS_2, ..., DS_n}
I5 = {DC_1, DC_2, ..., DC_n}
I6 = {O_1, O_2, ..., O_n}
I10 = Function “strcmp”
I11= Function “strsplit”
Outputs: DST_X vector [DS_x, DC_x, O_x]

$DSTXVEC \leftarrow strsplit(DSTX)$;
if $(strcmp\ (DSTXVEC(1),'\ word\ 2\ '))$
  $DCX \leftarrow DC1$;
  $OX \leftarrow \{O1, O2\}$;
else
  $DCX \leftarrow DC2$;
  $OX \leftarrow \{O6, O8, O10\}$;
end
for $idx1 \leftarrow 2 : length\ (DSTXVEC)$
  if $(strcmp\ (DSTXVEC(idx1),'\ word\ 3\ ' == 1)$
    $DSX \leftarrow DS1$;
  end
  if $(strcmp\ (DSTXVEC(idx1),'\ word\ 4\ ' == 1)$
    $DSX \leftarrow DS2$;
  end
  if $(strcmp\ (DSTXVEC(idx1),'\ word\ n\ ' == 1)$
    $DSX \leftarrow DSn$;
  end
end
Return $DSTXVEC$

To realize sub-function $F_{SA1,3}$, as a first step of the procedure, a matrix VectorTools need to be built for all DATs, as it is done by Algorithm A_{23}. To illustrate how this algorithm will work in a real-life situation, we introduced examples of inputs characterization in Line 3.

Algorithm 23. Build vector DAT

Inputs: I4 = {DS_1, DS_2, ..., DS_n}
I5 = {DC_1, DC_2, ..., DC_n}
I6 = {O_1, O_2, ..., O_n}
I7 = { DAT_1, DAT_2, ..., DAT_n }
Outputs: DAT_X vector [DS_x, DC_x, O_x]

1: **for** $idx1 \leftarrow 1 : length\ (I7)$
2:   **if** $(idx1 == 1 || idx == 4 || idx1 == 8)$
3:     $DAT \leftarrow [DS2\ \ DS4\ \ DC1]$;
4:   **else**
5:     $DAT \leftarrow [DS1\ \ DS3\ \ DS5\ \ DS6\ \ DC1\ \ DC2]$;

```
      end
      if (idx == 1)
          DAT ← [DAT O5];
    else if (idx == 4)
          DAT ← [DAT O4];
      else
          DAT ← [DAT O1];
      end
      VectorTools {idx1} ← DAT
  end
  Return DAT
```

As a second step, Algorithm A_{24} calculates the distance between vectors DST and DAT. Towards this end, it uses two functions: (i) "strfind," which determines the patterns within strings, and (ii) "isempty," which checks whether a string is empty, or not.

Algorithm 24. Calculate the distance between DSTs and DAT vectors

Inputs:	I12 = DAT_X vector I13 = DST_X vector I14 = Function "strfind" I15 = Function "isempty"
Outputs:	Distance between DSTX and DATX

```
for idx ← 1 : length (VectorTools)
    distance ← 0;
    for idx2 ← 1: length (DSTVECTOR)
        distance ← distance + isempty (strfind
        (char (VectorTools (idx1)), DSTVECTOR (idx2)));
    end
    distanceVector(idx1) ← distance;
end
Return distance
```

The third step is to sort DATs included in I_7 according to the distance to DAT (Algorithm A_{25}). This algorithm is eventually a simple "bubble sort" sorting algorithm (described in Section 5.3.2), which steps through a list, compares adjacent elements, and swaps them if they are in an incorrect order.

Algorithm 25. Sorting DATs

Inputs:	I7 = { DAT_1, DAT_2, ..., DAT_n }
Outputs:	DATX sorted by distance

```
for idx ← 1 : length (I7) : −1 : 1
    for idx2 ← 2 : idx1
        if (distanceVector (idx2 − 1) > distanceVector (idx2))
           tmp ← I7 (idx2 − 1);
           I7 (idx2 − 1) ← I7(idx2);
           I7(idx2) ← tmp;
```

```
            tmpd ← distanceVector(idx2 − 1);
            distanceVector (idx2 − 1) ← distanceVector(idx2);
            distanceVector (idx2) ← tmpd;
        end
    end
end
Return distanceVector
```

Once the distance has been calculated, all DATs, which fulfill the minimum distance criterion are retrieved. The weights assigned to the tools (I_8) are also retrieved and used later on. The computational details are specified in Algorithm A_{26}.

Algorithm 26. Retrieve the DAT vectors most similar to DST vector

Inputs: I8 = {W_1, W_2, ..., W_n}
I7 = {DAT_1, DAT_2, ..., DAT_n}
I16: distanceVector

Outputs: DATs with equal minimum distance to DST

```
distance ← distanceVector (1);
DATs ← [I7 (1)];
weightsSimilarVector ← [I8 (1)];
i ← 2;
While (distance == distanceVector (i))
        DATs ← [DATs I7(i)];
        weightsSimilarVector ← [weightsSimilarVector I8(i)];
        i ← i + 1;
        if (i > length(distanceVector))
          break;
        end
end
Return DATs
```

In order to rank the tools, which are at a minimum distance to DAT, a matrix of weights is used. Every tool has three different weights, one for each criterion (C_x). First, the algorithm produces the sum of weights for every tool, then it sorts the obtained sums in a descending order (RW stands for the ranked weights). Correspondingly, the tools are also sorted (RT stands for the ranked tools). The whole procedure is realized by Algorithm A_{27}.

Algorithm 27. Ranking DATs

Inputs: I8 = {W_1, W_2, ..., W_n}
I7 = {DAT_1, DAT_2, ..., DAT_n}
I17 = DATs with equal minimum distance to DST

Outputs: DATs sorted from high to low

```
for idx1 ← 1 : length (weightsSimilarVector)
    W ← char(weightsSimilarVector(idx));
```

```
        WC1 ← W(1);
        WC2 ← W(2);
        WC3 ← W(3);
        somWC1 (idx1) ← WC1 + WC2 + WC3;
    end
    RW ← sort(somWC1,'descend');
    [c,d] ← sort (somW1,‘descend’);
    RT ← [];
    for i ← 1: length(somWC1)
            RT ← [RT,Tis(d(i))];
    end
    Return RT
```

The final step is to select the tools of maximum sum and to present the final set of the best matching tools to the designer. The function used for this purpose is the “maxSom,” which ranks the sum of the weights and determines the one with maximum value (MW). The computational logic of the considered algorithm, Algorithm A_{28}, is presented below.

Algorithm 28. Retrieve best finite set of DATs

Inputs: I18 = Sum of weights
I8 = {W_1, W_2, ..., W_n}
I7 = {DAT_1, DAT_2, ..., DAT_n}

Outputs: Matrix of DATs with high weights

```
    MW ← max(somWC1(1));
    FinalMatrixTi ← [];
    for i ← 1: length(MW)
        FinalMatrixTi(i) ← [FinalMatrixTi,Tis(d(i))];
    end
```

Based on the specification of the algorithms and the decomposition of the recommendation function, the relationship between the algorithms can be determined. Figure 5.3 shows the computational workflow, as well as the communication among the algorithms and the procedural conditions. The algorithms were adapted in the same order, including their input–output relations that are necessary for the realization of the computational workflow.

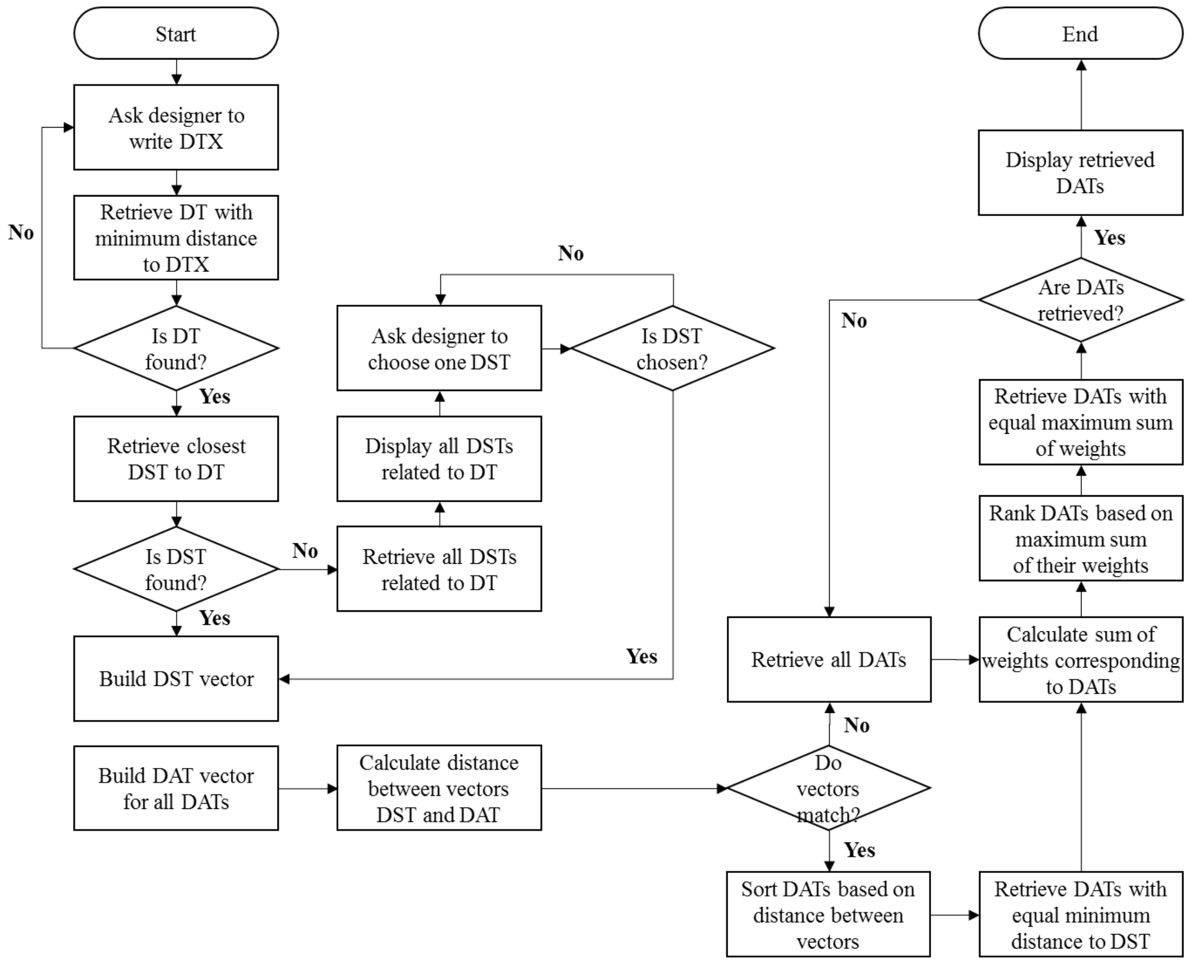


Figure 5.3. The computational workflow of the task-relevant data analytics tools recommendation module

5.4.3. Implementation of the smart user identification module

Most publications dealing with user identification using face authentication identify six main steps of the process: (i) accessing image source, (ii) face detection, (iii) face normalization, (iv) features extraction, (v) features matching, and (vi) identity validation [127-129]. Since these steps have been addressed with dedicated algorithms, below we consider only those algorithms presented in Table 5.3, which have been adapted or specifically developed from scratch for the purpose of smart user identification.

Algorithm A_{29} acquires a new image from the camera. This algorithm was not developed, since the software tools currently offered by Matlab allow achievement of this objective. For example, the related Matlab software allows accessing an image source and acquiring an image from it using the "image acquisition toolbox." In case of usage of other systems or platforms, Algorithm $_{29}$ (as presented in Table 5.3) has to be developed. Concerning face detection, we used the technique of weak classifiers characterization offered by the AdaBoost algorithm, as explained in Section 5.3.3. If a set of training data (O_1,y_1), …, (O_L,y_L) is considered, where $y_i \in \{-1,+1\}$ assigns the example O_i to the class -1 or +1, and the number of weak classifiers to be found is M, then the AdaBoost algorithm is as follows:

Algorithm 30. Detect a face in an image (AdaBoost)

Inputs: I1 = Family of weak classifiers: $\{C_\omega\}$
I2 = Procedure $getBWC$: $[\omega, \epsilon] \leftarrow getBWC(\dots)$
I3 = Procedure $getGDC$: $[\alpha, \beta] \leftarrow getGDC(\dots)$

Outputs: Costs of positive hypothesis: $\alpha_1, \dots, \alpha_M$
Costs of positive hypothesis: $\beta_1, \dots, \beta_M$

for $i = 1, \dots, L: \; w_{i,1} \leftarrow 1/L$; % Initialize weights
  for m = 1, …, M :
    $[\omega_m, y'_1, \dots, y'_L, e_t] \leftarrow getBWC(O_1, \dots, O_L;\; w_{1,m}, \dots, w_{L,m};\; y_1, \dots, y_L)$; %
    Select the optimal classifier, its hypotheses and error
    $[\alpha_m, \beta_m] \leftarrow getGDC(w_{1,m}, \dots, w_{L,m};\; y_1, \dots, y_L;\; y'_1, \dots, y'_L)$; % Select
    costs of positive and negative hypotheses
  for $i = 1, \dots, L: \; w_{i,m+1} \leftarrow w_{i,m} e^{-\gamma_t(O_i) y_i}$; % Update object weights
    $Z_t \leftarrow 0$; % Normalize weights
    for $i = 1, \dots, L: \; Z_m \leftarrow Z_m + w_{i,m+1}$;
    for $i = 1, \dots, L: \; w_{i,m+1} \leftarrow w_{i,m+1}/Z_m$;
Return $[\omega_1, \dots, \omega_M, \alpha_1, \dots, \alpha_M, \beta_1, \dots, \beta_M]$

In the process of finding a face in an image, we used the principle of extracting sub-image s from given image f, to detect a face. The sub-image is of size m × n and the coordinates of its top left corner are (r_x, c_y). The corresponding algorithm is presented below:

Algorithm 31. Detect face in image

Inputs: I1 = Function s=subim(f, m, n, rx, cy)
I2 = Image "f"

Outputs: Sub-image "s" of coordinates (x_{count}, y_{count})

$s = zeros(m, n);$
$rowhigh \leftarrow r_x + m - 1;$
$colhigh \leftarrow c_y + n - 1;$
$x_{count} \leftarrow 0;$
for $r \leftarrow r_x : rowhigh$
  $x_{count} \leftarrow x_{count} + 1;$
  $y_{count} \leftarrow 0;$
  for $c \leftarrow c_y : colhigh$
    $y_{count} \leftarrow y_{count} + 1;$
    $s(x_{count}, y_{count}) \leftarrow f(r, c);$
  end
end
Return $s(x_{count}, y_{count})$

Another adapted algorithm was the one used for face feature extraction. It used the principles of the edge descriptor. This is Algorithm A_{33}, presented below:

Algorithm 33. Extraction of face features (adapted edge histogram descriptor algorithm)

Inputs: I1 = function H=ehd(Img, Threshold)

I2 = original image “Img” of size (x_{old}, y_{old}, z)

Outputs: Histograms representation

```
x ← x_old/4;
y ← y_old/4;
if mod(x,2)~ ← 0
   x ← x + 1;
end
if mod(y,2)~ ← 0 0
   y ← y + 1;
end
Img_2 ← unit8(zeros([x * 4   y * 4   z]));
Img_2(1:x_old,  1:y_old,   1:z) ← Img(1:x_old,  1:y_old,   1:z);
Img ← Img_2;
Image ← rgb2gray(im2double(Img));
H ← [ ];
  HorizontalMask ← [1 1 1;0 0 0;−1 −1 −1];
  VerticalMask ← [1 0 −1;1 0 −1;1 0 −1];
  DiagonalMask ← [0 1 1;−1 0 1;−1 −1 0];
  AntiDiagMask ← [1 1 0;1 0 −1;0 −1 −1];
HImage ← imfilter(Image,HorizontalMask,'replicate');
VImage ← imfilter(Image,VerticalMask,'replicate');
DImage ← imfilter(Image,DiagonalMask,'replicate');
AImage ← imfilter(Image,AntiDiagMask,'replicate'); %NImage←imfilter (Image,DirectionalMask, ‘replicate’) determines the size of each sub-image dynamically
m ← size(Image,1)/4; % Number of rows per sub-image
n ← size(Image,2)/4; % Number of columns per sub-image
column ← 1;row ← 1;counter ← 1; % Initialize variables
for k ← 1:16 % Loop through every sub-image
    if counter > 4
      column ← 1;
      row ← row + m;
      counter ← 1;
    end
    subImgH(k).img ← subim(HImage,m,n,row,column);  % Get subimage for H
    subImgV(k).img ← subim(VImage,m,n,row,column);  % Get subimage for V
    subImgD(k).img ← subim(DImage,m,n,row,column);  % Get subimage for D
    subImgA(k).img ← subim(AImage,m,n,row,column);  % Get subimage for A
    column ← column + n;
    counter ← counter + 1;
end
```

```
    for k ← 1:16 % All sub-images
        HLocal(k,1:5) ← 0;
    row ← 1;
    column ← 1;
    for m ← 1:ceil(size(subImgH(k).img,1) ∗ size(subImgH(k).img,2)/
        4)  % All 2×2 blocks
        if column > size(subImgH(k).img,2)
            column ← 1;
            row ← row + 2;
        end
        [M,I] ← max([sum(sum(abs(subImgH(k).img(row:row + 1,
                    column: column+1))))/4 …
                    sum(sum(abs(subImgV(k).img(row:row + 1,column:
                    column + 1))))/4 …
                    sum(sum(abs(subImgD(k).img(row:row + 1,column:
                    column + 1))))/4 …
                    sum(sum(abs(subImgA(k).img(row:row + 1,column:
                    column + 1))))/4 … Threshold]); % Determine the maximum
                    edge of the averages in 2×2
        IndexedSub(k).img((row + 1)/2,(column + 1)/2) ← I;  % Used
        for displaying edges
        HLocal(k,I) ← HLocal(k,I) + 1;
        column ← column + 2;
    end
    HLocal(k,:) ← HLocal(k,:)./(ceil(size(subImgH(k).img,1) ∗ size
    (subImgH(k).img,2)/4));
end
for k ← 1:16
    H ← cat(2,H,HLocal(k,:)); % ACreate main edge histogram to return
end
for k ← 1:4
    H ← cat(2,H,sum(HLocal(k:4:16,:))./4); % Add vertical groups
end
for k ← 0:3
    H ← cat(2,H,sum(HLocal(k ∗ 4 + 1:k ∗ 4 + 4,:))./4); % Add Horizontal
    Groups
end
H ← cat(2,H,sum(HLocal([1 2 5 6],:))./4); % Add neighbor groups
H ← cat(2,H,sum(HLocal([3 4 7 8],:))./4);
H ← cat(2,H,sum(HLocal([9 10 13 14],:))./4);
H ← cat(2,H,sum(HLocal([11 12 15 16],:))./4);
H ← cat(2,H,sum(HLocal([6 7 10 11],:))./4);
H ← cat(2,H,sum(HLocal(:,:))./16); % Add global group
Return Figure bar (H)
```

The histogram is expected to be in the order of local, neighbor, or global. In order to match face features, Algorithm A_{34} was developed. It calculates the distance between two EHD histograms, as presented below.

Algorithm 34. Calculate distance between two EHD histograms

Inputs: I1 = EHD of the first image "a"
I2 = EHD of the second image b"
I3 = Local weight "l"
I4 = Neighbor weight "n"
I5 = Global weight "g"
I6 = Function to calculate the distance [dist] = ehddist(a, b, l, n, g)

Outputs: Distance between two EHD histograms

$dist = 0;$
for $i \leftarrow 1{:}80$
    $dist \leftarrow dist + l * abs(a(i) - b(i));$
end
for $i \leftarrow 81{:}145$
    $dist \leftarrow dist + n * abs(a(i) - b(i));$
for $i \leftarrow 146{:}150$
    $dist \leftarrow dist + g * abs(a(i) - b(i));$
end
Return $ehddist(a, b, l, n, g)$

To identify the user and validate his identity, Algorithm A_{35} was designed to find similar identity, if it exists.

Algorithm 35. User identity validation

Inputs: I1 = EHD distance between histograms
I2 = features extracted
I3 = function "findsimilar(img)"

Outputs: Found similar identity

$n \leftarrow findsimilar(img)$
$similarities \leftarrow [];$
$load\ features$
$[csd128\ , edge] \leftarrow calcfeatures(img);$
for $k \leftarrow 1{:}size(names, 1)$
    $b \leftarrow pdist([csd128;\ csd128hist(k,:)]);$
    $d \leftarrow ehddist(edge, edges(k,:), 1, 1, 5);$
    $similarities \leftarrow [similarities;\ b\ d];$
end
for $k \leftarrow 1{:}size(similarities, 2)$
    $m \leftarrow mean(similarities(:, k));$
    $s \leftarrow std(similarities(:, k));$
    $similarities(:, k) \leftarrow (similarities(:, k) - m)\ ./\ s;$
end
$similarities(:, 2) \leftarrow (0.8.* similarities(2));$
$sims \leftarrow sum(similarities');$

17: $[\sim, I] \leftarrow sort(sims);$
18: $n \leftarrow names(I(:), :);$
19: **end**

The other algorithms, which are not new, were extracted from existing publications [130-134] and used in the particular case of the smart user identification module.

Based on the description and the principles of the smart user identification module and its developed algorithms, the computational workflow of this module is presented in Figure 5.4. After the image is accessed using the "image acquisition toolbox" in Matlab, it is captured and preprocessed (color conversion, normalization, noise removal, and classification) to allow the detection of the designer's face. Once the face is detected, it is registered in the database. The features are then extracted from it (using the dedicated algorithm presented above), and it is compared to database images by reasoning with existing face images and then matched with the designer's existing saved images. This new face capture is added to the database of the designer and saved into his or her attendance document (including the time and date of the log in). This final identification gives the designer access to the SDATB to start analysis. If more than one image is detected, the module allows a return to the image acquisition toolbox to recapture the image. When a new designer is added, the captured face is entered in the database under a new user, and a new attendance form is created for the designer.

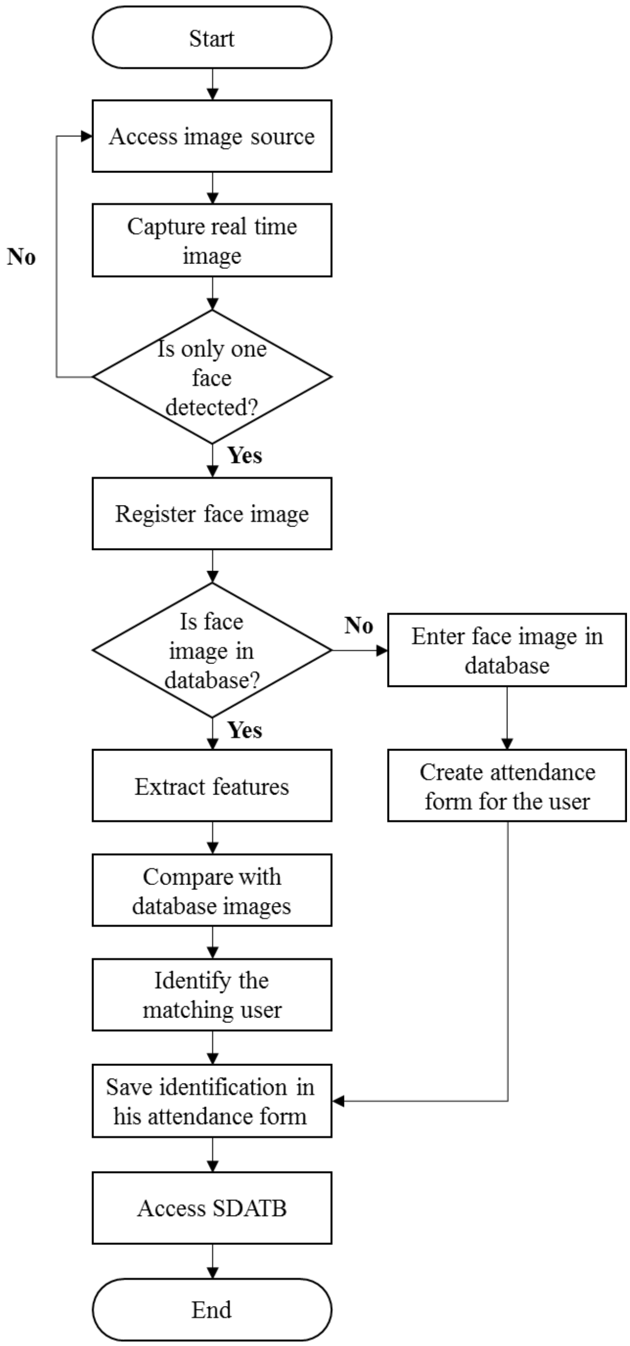


Figure 5.4. Workflow diagram of the smart user identification module

5.5. Functionality validation of the demonstrative data analytics toolbox modules

The objective of functionality testing was to determine (i) if the modules and their corresponding algorithms are operating as they are supposed to and (ii) to what extent the functional requirements have been fulfilled. In our case we consider three modules: (i) the merging MoLD-Ss recommendation module, (ii) the task-relevant data analytics tools module, and (iii) the smart user identification module. Functionality testing for each module is discussed in a separate section. For validation, the implemented modules have been considered in a particular application context related to white goods. This interest came because the white goods category covers a large field of products. These products are heavy consumer durables that include all home appliances related to refrigeration, cooking, washing, drying, heating, and cooking. Moreover, these products are equipped with advanced control units and relatively high numbers of sensors able to collect MoLD. The majority of them are known by the continuous evolution towards smart products. Accordingly, the demonstrative SDATB modules were tested as part of a process by white goods designers of enhancing a particular connected washing machine.

5.5.1. Testing the functionality of the merging middle-of-life data streams module

To computationally implement the merging MoLD-Ss module and test its functionality, we used our reference application case of a connected washing machine. Accordingly, we considered that this device has 13 sensors (S_x where x is the sensor's number), represented as follows:

- S_1: Force gauge on the axle bearings of the washing drum. This sensor communicates the speed of the washing drum as well as the force applied on it.
- S_2: Force gauge on transmission belt. This sensor communicates the speed of the transmission belt and the force applied on it.
- S_3: Brake shoes position sensor. This sensor communicates the position of the brake of the washing drum and how much it brakes.
- S_4: Force gauge on brake spring. This sensor communicates the force applied to the spring connected to the brake.
- S_5: Spinner time control clock. This sensor communicates how long the washing machine was set to spin for, how long it has been spinning, and the time left until it is done.
- S_6: Washing timer control clock. This sensor communicates how long the machine was set to wash for, how long it has been washing, and the time remaining until it is finished.
- S_7: Detector of spinning R.P.M. setting. This sensor communicates the speed of the spinning drum.

- S_8: Water level indicator. This sensor communicates the amount of water needed for the washing and how much water was actually used.
- S_9: Inside temperature sensor in the housing. This sensor communicates the external temperature of the room where the washing machine is installed.
- S_{10}: Solid deposition indicator in the outlet of the wastewater pipe. This sensor communicates the condition of the water pipe. It indicates if it is clean and working perfectly or is blocked by dirt.
- S_{11}: Switch on/off detector counter. This sensor communicates when the washing machine is on and when it is off. This sensor activates all the other sensors.
- S_{12}: Heater temperature thermometer. This sensor communicates the variation of temperature when the machine is in use.
- S_{13}: Heating time counter. This sensor communicates the time needed by the washing machine to heat and achieve the needed temperature.

Since we do not have access to real data streams, we built fake data streams (some streams have anomalies, others do not). In addition, we incorporated prior knowledge for product anomalies in the data streams. In total, five different failures were described with sensors, and actions were recommended for design changes and enhancement.

We introduce a term of causality matrix C, where $C_{ij} = 1$ if the i_{th} anomaly can happen in the event of a failure, and the j_{th} sensor is capable of capturing the device part dysfunction causing that anomaly. In general, sensors can describe dysfunction in different device parts. Yet, all of them are semantically linked to the same anomalous behavior (e.g. electrical circuit failure or mechanical parts being worn out).

If a pair of anomalies i and sensors j have no causal relationship, then $C_{ij} = 0$. Normal behavior is represented with a row C_i in which each value is equal to 0. Each sensor has its semantic meaning, which affects its periodicity, its values range, and the type of time series that can be present in its data stream. In this sense, a unique function has been developed to mimic sensor's activity.

For the sake of the functionality testing, five anomalies (An_x, where x is the anomaly number) and their possible action plans were built and described. We created a mapping between the anomalies, related sensors, and recommendation messages. The mapping is presented in Table 5.4.

Regardless of the anomaly type, if a particular sensor must exhibit a faulty signal, we manually engineer anomalous sensor activity. A single data stream is represented by a function *f(t)*, such as

$$f(t) = f(t \mid j, c, \eta), \tag{5.21}$$

where j is sensor number, c indicates whether an activity is normal or faulty, η is a source of randomness to have multiple instances per (t, j, c). In the simplest case, we randomly shift a periodic function to guarantee that values vary across instances of the data stream. For testing purposes, we generated more complex but consistent MoLD-Ss. The logic behind data stream generation is presented in Table 5.5. It represents the difference between the normal behavior of sensors and faulty behavior. Consequently, the

Table 5.4. Mappings between anomalies, sensors, and recommendation messages

Anomaly code	Description	Related sensors	Recommended action
An_1	Mechanical wear out of most-used components in the washing machine (washing drum, brakes to stop the drum, and related components).	S_1 or S_2 or S_3 or S_4	Mechanical control, adjustment, or replacement of components is needed
An_2	Incorrect values for typical washing cycle parameters, such as water volume, temperature of heating element and water, amount of time for washing and drying, and speed of drum rotation.	S_5 or S_6 or S_7 or S_8 or S_9	Electronic sensor control, adjustment, or replacement of components is needed
An_3	Corroded wires, waste pollution, or other chemical or unexpected substances aggregating in the washing machine that can cause either general power supply issues or locally block or slow down flow of water inside the washing machine.	S_{10} and S_{11}	Mechanical and chemical cleaning are needed
An_4	Deviation in mechanical and electrical components' sensor values, implies that a general setup is incorrect. It can be a tilted washing machine, a plug removed from the power supply socket, or incorrect device assembly after previous repairs. A cause that does not allow the device to start or shuts it down after start due to diagnostics failure.	(S_1 or S_2 or S_3 or S_4) and (S_5 or S_6 or S_7 or S_8 or S_9) and S_{11}	This appliance is incorrectly selected or installed for this application
An_5	Abnormal temperature values and heating time deviation, with potentially sporadic device terminations. This can be caused by overheating or under-heating issues.	S_{11}, S_{12}, S_{13}	Water heater element should be cleaned or replaced

interpretation of MoLD-Ss is easier. All visual representations of normal and faulty behaviors of sensors can be found in Appendix 5.

We may derive novel anomalies by mutating a matrix C_j such that

$$\sum_{j=1}^{M} |C'_{ij} - C_{ij}\ | < K, \tag{5.22}$$

where M is the number of sensors, and K represents the number of introduced novel sensor dependencies. In the testing process, we add only one additional dependency to each case to obtain a test set of anomalies.

Table 5.5. Normal and faulty behaviors for each of the thirteen sensors

Sensor code	Normal behavior	Faulty behavior
S_1	Constant force during the whole washing cycle.	Abnormal force at some moments during the washing cycle.
S_2	Constant force during the whole washing cycle, with a greater deviation than washing drum force gauge axle.	Abnormal force at some moments during the washing cycle, correlated with S_1 faulty activity.
S_3	Rapid application of brake before the end of washing cycle, with a constant position of brake shoe.	The position is not constant and fluctuates within a small margin.
S_4	Steadily increasing force during the brake application.	A large Gaussian noise is added to the force value. It models a loose contact between the brake shoe and the surface.
S_5	Regular positive voltage that indicates spinning operation.	Irregular positive voltage for spinning indication. It models a problem with drum rotation.
S_6	Less frequent than S_5, but still regular positive voltage that indicates a change in the washing stage.	Irregular positive voltage for the washing stage change. It models incorrect washing process (longer high-temperature periods, rinsing ignored).
S_7	A regular 3-stage spinning operation, with a slow rotation, a quicker run for the main stage, and again a slower rotation.	More rotation speed changes, with switches between very slow modes and very fast modes or vice versa. It models broken drum speed control.
S_8	Increase in the water level until a target level is reached, which is 3, 4, or 5 liters. At the end of the cycle, water is drained rapidly. It is repeated 2 times. The water level is the same for both iterations.	More than 2 water change cycles with different water levels. It might lead to water being present during the drying or absence of detergent at later stages.
S_9	Steady temperature increase with a constant temperature during washing and the temperature decreasing during rinsing stage.	Multiple temperature change cycles with different temperature settings during a washing cycle. It might indicate a heating element issue or just a microcontroller failure to follow a preprogrammed behavior.
S_{10}	Gaussian noise around 0 voltage to indicate absence of solid waste in the outlet.	Random nonzero voltage indicating the presence of a blockage in the outlet. It models bad water circulation due to a reduced flow capacity.
S_{11}	Single positive voltage when machine is turned on and a single negative voltage when machine is turned off at the end of the washing cycle.	Multiple switches between on and off before the machine turns on robustly. It models issues with a device start and can be caused by an electrical circuits issue or general diagnostics failure or the device not being correctly installed.
S_{12}	Heater temperature is increasing or decreasing until a certain room	Many more changes in the temperature. If the room temperature sensor is faulty, then these

	temperature level is reached. Due to transmission time, there is a lag before the heater changes its state.	two are correlated. Otherwise, it should be considered a model of sensor fault rather than a heating element. Or the room temperature may not be affected due to conduction issues.
S_{13}	Same as in S_{11}: a positive voltage at the start of heating state change and a negative voltage at the end of it.	More changes in heating states.

A dataset was used to train the neural network architecture. As mentioned in Section 5.4.1, we used SGD training with mini-batches, and the loss function is a triplet loss. Our model implementation assumes that it operates in a sliding window fashion along the time axis to encode behavior patterns and find the closest match among past cases. This is mathematically described as follows:

$$X_{i,j,t} = f(t \mid j, C_{ij}, \eta), \qquad (5.23)$$

$$h_k = g((X \mid k - {}^{T}/_{2} < t < k + {}^{T}/_{2})), \qquad (5.24)$$

where $X_{i,j,t}$ is a joint input matrix of raw data streams, h_k is a behavior descriptor of input data for a sliding window defined by time step k and window length T, and the function $g()$ is the neural network encoder of the device behavior pattern based on input data. It includes single-stream encoding, sensor importance estimation, and merging of multimodal data and representation as a fixed length vector. Since triplet loss requires three multimodal data streams, we randomly generate B triplets, such that $i_{l,1} = i_{l,2}$ and $i_{l,1} \neq i_{l,3}$, where $i_{l,1}$, $i_{l,2}$, and $i_{l,3}$ are anomaly types in the triplet; $t_{l,i}$ is the time position of windows; T is the window length; and $i_{l,p}$ and $t_{l,p}$ are random.

During model optimization we have multiple hyperparameters to optimize. These include the learning rate Ir, the batch size B, the window length T, the size of intermediate representation for single data stream L_1, the size of latent representation before passing to attention layer L_2, and the size of latent representation for behavior L_3. The neural network encoder can be disassembled into multiple components presented in expression 5.25:

$$a = g_1(X)\,, a \in R^{B\times3\times L_1}, b = g_2(a), b \in R^{B\times3\times L_2}, c = g_3(b), \qquad (5.25)$$
$$c \in R^{L_2}, h = g(X) = g_4(b, c), h \in R^{B\times3\times L_3},$$

where a is the output from the first layer of neural networks ($g_1 = conv_1$), b is the second layer ($g_2 = conv_2$), c is the attention layer ($g_3 = attention$), and d is the final behavior descriptor ($g_4 = behavior.conv_1$). During the architecture design, we experimented with the elimination of the $conv_2$ layer to compare a deeper model with a shallow one.

As stated in Section 5.3.1, data normalization was used for preprocessing in some of our tested architectures. To train the model, we considered $Ir = 1e - 3$, $B = 32$, $T = 256$, $C \in R^{6\times13}$, $L_1 = 16$, $L_2 = 16$, and $L_3 = 16$. The optimization objective was to minimize triplet loss, and as a matrix we used a separation accuracy by a margin β. Figure 5.5 presents five experiments we conducted and their learning curves. Below is the interpretation of this figure:

- “Deep, not normalized” is an architecture with $conv_2$ present, but the input data were not normalized. The model was trained for five epochs, each with 128 SGD steps.
- “Shallow, not normalized” is an architecture without the $conv_2$ layer. The number of epochs and epoch size were the same as in the previous model.
- “Shallow, not normalized, v2” is the same architecture as the previous model but trained for more steps. The number of epochs is 10, and the epoch size is 512.
- “Shallow, normalized” is the same experiment as the previous one, but the input data were preprocessed with normalization.
- “Deep, normalized” is the architecture from the first experiment “deep, not normalized,” but the input was preprocessed with normalization. The number of steps was greater, as in the case of “shallow, not normalized”: 10 epochs with 512 steps each.

The analysis of the obtained results revealed that, without data normalization, the deep model was trained to 70% accuracy within the first 128 steps but diverged afterwards because the learning rate was not gradually decreased but was fixed at $1e - 3$. It is not clear whether this model would train towards less error or the learning rate would decrease every time the optimization does not improve for several consecutive steps. Data normalization produced much more stable learning and achieved better results, with an accuracy of 83% and the smallest loss values across the five presented architectures. The shallow models with and without normalization did not perform well. Either a minimum with a larger error was reached much more quickly, and the training did not progress within the remaining five to six epochs, or a model did not converge at all for the first few epochs and rapidly reached a similar error to the best one.

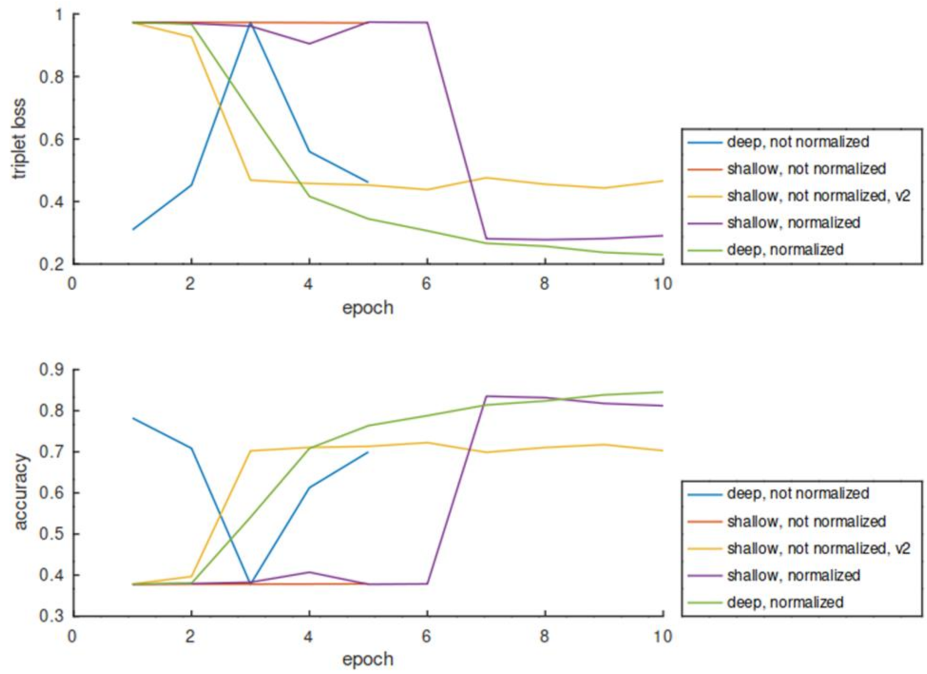


Figure 5.5. Learning curves of conducted experiments

In the implementation of the proposed computational function, we used not only triplet loss as an objective but also included an attention layer in the architecture to verify whether the relevant sensors would get higher fusion weights than the other ones. The attention layer allows reweighting of the latent representations of single streams before passing them to further layers of the neural network. In our experiments, we observed that the model did not differentiate between fewer data streams of faulty cases and multitude of normal ones, because in practice the model assigns smaller weights to faulty sensors. We ordered sensors based on attention weights and pass forward couple of sensors with the smallest values, as well as several sensors with the highest values. This procedure guarantees that despite the way attention layers were trained, both cases will be handled.

We performed a database search for matching behavior descriptors and the initial detection of anomalies with comparisons to all six cases: five anomalies and one normal behavior. In the dataset training, we used the original causality matrix $C_{i,j}$ and performed the analysis on a mutated causality matrix C'_{ij}. If the accuracy was above the threshold, then those pairs were selected. During anomaly detection, we looked for the pairs that were most similar and retrieved the closest match and the identifiers of the most important sensors based on the attention layers' prediction. These pieces of information are used for generating recommendations.

To test the functionality of the merging MoLD-Ss recommendation module, we considered the application case presented in the first paragraph of this section, as we implemented the mentioned reasoning and learning procedures as hidden operations behind a graphical user interface (GUI) developed in Matlab. We adopted the definition that refers to it as a software platform designed with visual components (icons, windows, menus, etc.) allowing a user to easily navigate and interact with inputs and outputs requirements [71]. We decided to implement a simple GUI to visualize this module for the designer from his or her point of view (of course the interface of the actual data analytics toolbox will be much more sophisticated).

The main screen of this module, as presented in Figure 5.6, includes two actions: (i) "Data" containing one option called "Select Sensors" for choosing which sensors to analyze, since our sensors are already located in the platform, and (ii) "About," which displays general information about the function. A designer who clicks on "Select

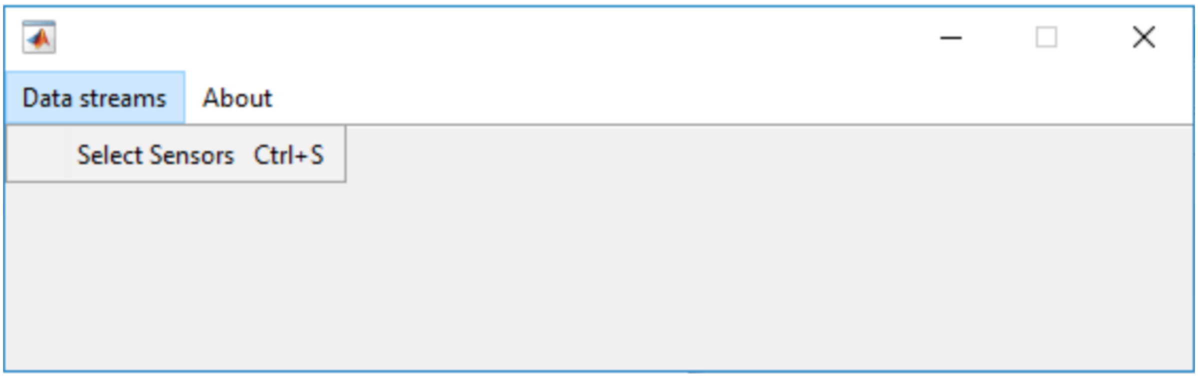


Figure 5.6. Main screen of the merging of middle-of-life data streams module

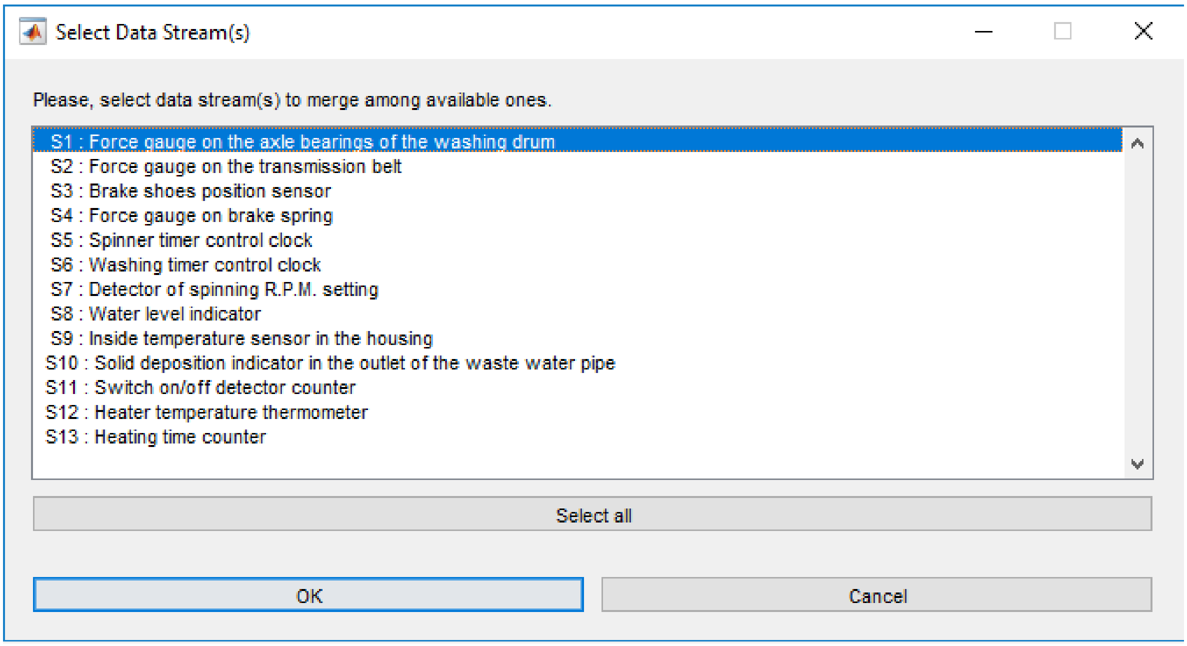


Figure 5.7. "Select Data Stream(s)" screen of the merging of middle-of-life data streams module

Sensors" is moved to the next screen, which displays available MoLD-Ss with their corresponding codes and a short description of each.

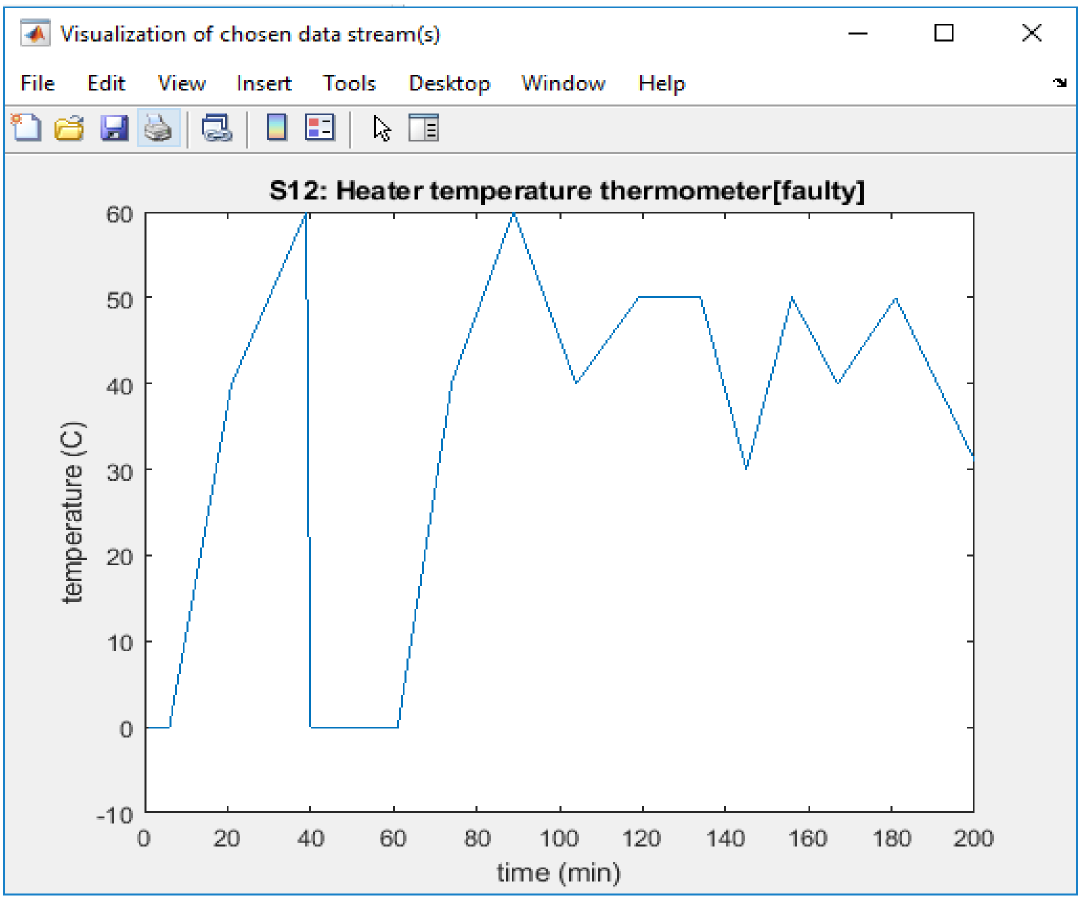


Figure 5.8. Visualization of sensor 12

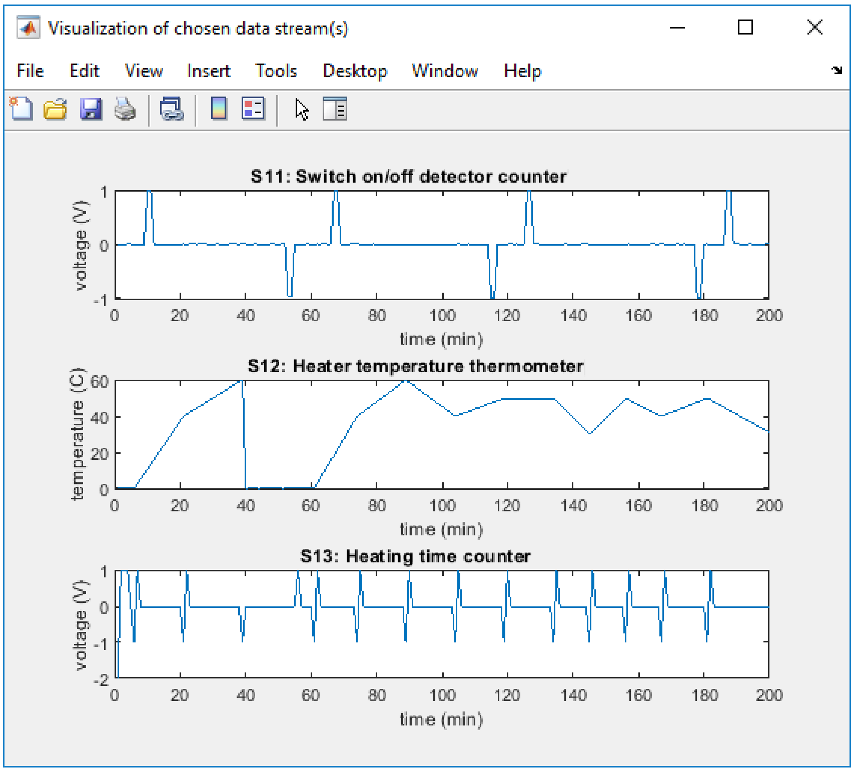


Figure 5.9. Visualization of combined sensors 11, 12, and 13

At this level, the designer chooses which sensors to merge (the option "Select all" is also available), or chooses one sensor if he or she only wants to analyze a particular sensor, and then presses "OK" to continue with the visualization or "Cancel" to return to the initial screen (Figure 5.7). After the designer's choice (we assume that the designer selects S_{11}, S_{12}, and S_{13}), the MoLD-Ss are transferred to be analyzed. The following screen is called "Visualization." Once the inspection of represented plots is completed, the designer needs to press "×" to return to the previous window to select the button to merge data streams, as presented in Figure 5.10.

The merging is performed in the background of the GUI. The designer is only presented with a final textual recommendation within seconds. The recommendation message contains an explanation of detected anomalies and their sensors, as well as the recommendation (or action plan) semantically related to anomalies from different sensors. The message communicated based on the assumed choices presented above is displayed in Figure 5.11. As can be seen in this figure, S_{11} is not mentioned in the

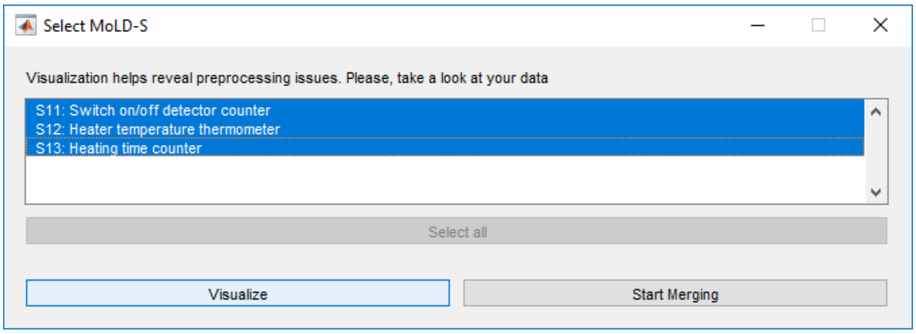


Figure 5.10. Window for selecting "Start Merging"

message. This means that no anomalous behavior was detected related to that specific sensor, but its semantic meaning was used in delivering the recommendation. If the user of the washing machine had turned on and off the device more often than, S_{11} would have reflected that and consequently serious measures and different recommendation would be advices. Perhaps not only the water heater element is an issue but the whole electrical system of the machine is failing. To check the relevance of the analyses, we repeated the merging three times for the same sensors, and we obtained the same anomalies with the same recommendation.

To sum up, the functionality testing proved that the objective set for this module was achieved. From a computational point of view, the algorithms designed for this function and the ones taken from the literature were able to be converted, showing no computational errors. Based on the results shown in Figure 5.24, it was observed that the reasoning and learning from MoLD-Ss played a significant role in the formulation of the recommendation message to be delivered to the designer. The message covers not only the detected anomalies but also recommends certain actions to be considered by the designer. We concluded that, at the beginning of the implementation, the conditions set for the conversion of faulty behaviors of the MoLD-Ss into a concrete action plan for the designer were correctly elaborated. The application of the merging MoLD-Ss function (i) provides more information than can be obtained by processing the sensors' data individually, (ii) reflects the condition of the product, (iii) communicates information about the product while it is in use by the customer, (iv) reduces sensor analysis time and effort, and (v) provides a recommendation as an action plan for the product at hand. Offering this function to product designers will allow them to continually analyze their products' behaviors and quickly develop enhancements and solutions.

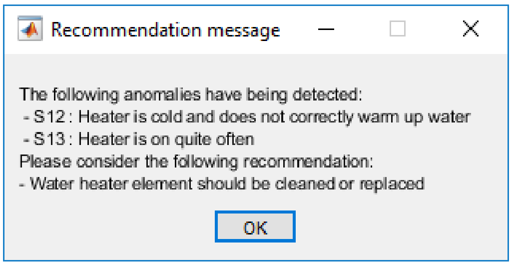


Figure 5.11. Recommendation message communicated to the designer

5.5.2. Testing the functionality of the task-relevant data analytics tools recommendation module

To test the functionality of the task-relevant data analytics tools recommendation module, it was computationally implemented in Matlab using the application case of recommending the appropriate DATs for a designer in the process of enhancing a particular connected washing machine using the SDATB. In this sense, the first step consisted of specifying the global inputs of the module. Consequently, we identified a set of DTs (input I_1) and their corresponding DSTs (I_2). For the sake of simplification, three DT_x were used, and their corresponding DSTs were presented as $DT_{x,y}$, where x is the code of each main DT, and y represents the order of appearance of DSTs. Below is the representation of the elements of I_1 and each corresponding I_2:

- DT_1: Enhancement of product performance:
  - $DT_{1,1}$: Analyzing energy consumption
  - $DT_{1,2}$: Analyzing water consumption
  - $DT_{1,3}$: Analyzing temperature settings
  - $DT_{1,4}$: Analyzing loading
  - $DT_{1,5}$: Analyzing detergent usage
- DT_2: Enhancement of product design:
  - $DT_{2,1}$: Analyzing most used features
  - $DT_{2,2}$: Analyzing relationships between most used features
  - $DT_{2,3}$: Analyzing least used features
- DT_3: Enhancement of product lifecycle:
  - $DT_{3,1}$: Analyzing product components
  - $DT_{3,2}$: Scheduling of predictive maintenance
  - $DT_{3,3}$: Scheduling of preventive maintenance

The second step is the identification of the DSs of the washing machine (I_4). Seven data sources were identified:

- DS_1: Temperature sensor (for $DT_{1,3}$)
- DS_2: Water flow sensor (for $DT_{1,2}$)
- DS_3: Load sensor (for $DT_{1,4}$)
- DS_4: Detergent level sensor (for $DT_{1,1}$, $DT_{1,5}$)
- DS_5: Event log (for $DT_{2,1}$, $DT_{2,2}$, $DT_{2,3}$, $DT_{3,1}$)
- DS_6: Maintenance history (for $DT_{3,2}$)
- DS_7: Maintenance report (for $DT_{3,3}$)

The third step is the identification of possible DCs (I_5). For simplification, two main categories of data were identified:

- DC_1: Big data (coming from DS_1, DS_2, DS_3, DS_4, DS_5)
- DC_2: Small data (coming from DS_6, DS_7)

The fourth step is the identification of possible Os (I_6) based on data category. Since we were testing the functionality of the recommendation system, a limited number of outputs were used:

- O_1: Plots (possible for DC_1, DC_2)
- O_2: Hierarchical tree (possible for DC_1, DC_2)

- O_3: Dendrogram (possible for DC_2)
- O_4: Hyperplane (possible for DC_2)

The fifth step is the identification of DATs (I_7) that are included in the SDATB (referred to as T_x for simplification and to prevent later coding errors). For the purposes of demonstration, some machine learning tools from the "statistics and machine learning toolbox" of Matlab were considered, as listed below:

- T_1: Support vector machines (can analyze DC_2 and provide O_4)
- T_2: Decision trees (can analyze DC_1 and DC_2 and provide O_2)
- T_3: Classification trees (can analyze DC_1 and DC_2 and provide O_2)
- T_4: K-nearest neighbors (can analyze DC_2 and provide O_1)
- T_5: K-means (can analyze DC_1 and DC_2 and provide O_1)
- T_6: K-medoids (can analyze DC_1 and DC_2 and provide O_1)
- T_7: Hierarchical clustering (can analyze DC_2 and provide O_3)
- T_8: Gaussian mixture models (can analyze DC_2 and provide O_1)

The last step is the identification of the weight matrix for each of the tools according to C_1, C_2, and C_3. If our algorithm was a machine learning algorithm, the weights could be automatically defined. To avoid fundamental mistakes, it is important to mention that the weights were arbitrarily generated for the purpose of testing the recommendation function F_{SA1}. Below is the list of eight weight matrixes for the corresponding eight tools:

• W_1 = [3 10 5];	• W_5 = [7 5 5];
• W_2 = [2 7 4];	• W_6 = [7 4 7];
• W_3 = [8 3 1];	• W_7 = [10 1 1];
• W_4 = [1 6 8];	• W_8 = [1 3 6];

Figure 4.5 is revisited, adapted, and specialized to the application case, as shown in Figure 5.12. This figure was generated for the purpose of checking the validity of the recommendation module. On the left side of this figure, the acronyms of the tasks and subtasks of the designers, which were detailed previously, are listed. On the right side of the figure, the acronyms of the data analytics tools for testing are listed. In the middle of the figure is the matrix that matches designers' tasks and tools based on DS, DC, and DO criteria. If the design task described by the designer is close enough to a specific DT_x and $DT_{x,y}$ from the figure, then the DATs recommendable for the designer should be the ones that most correspond to DT_x and $DT_{x,y}$, indicated in the right list of Figure 5.12.

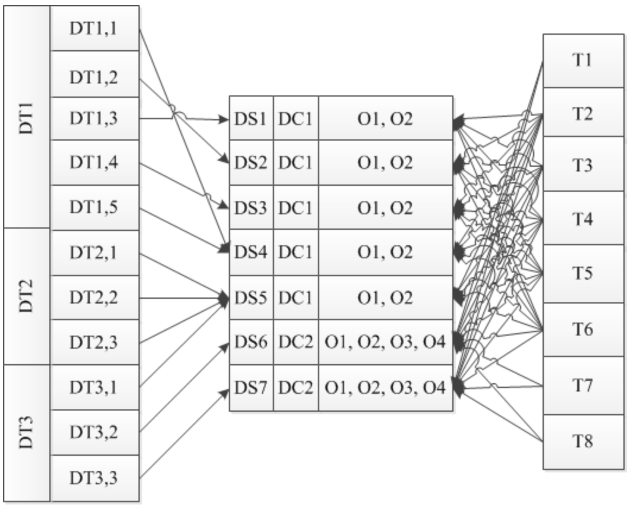


Figure 5.12. Adaptation of recommendation principle in a particular application case

After defining all global inputs needed for the realization of the recommendation module, we converted the algorithms detailed in Section 5.4.2 into pseudo-codes and inserted them as a new “Matlab script.” The lists of inputs were also included. Before we compiled the script, the functions used in the algorithms that are unknown to Matlab had to be defined (e.g. EditDistance). After programming the function, the designer can write down the design task and communicate it to the SDATB. We considered that the designer wanted to analyze a washing machine loading. This led to the following DT being communicated by the designer: DTX = “washing machine load.” After inserting DTX in the Matlab code and running it, we could recognize which DT*x* and DT*x*,*y* were closest to the designer’s DT. Consequently, the objective of $F_{SA1,1}$ was achieved. We found that the closest task and subtask were $DT_x = DT_1$ = “Enhancement of product performance” and $DT_{x,y} = DT_{1,4}$ = “Analyzing loading,” as shown in Figure 5.13. In this figure, ISB12 is I_2 (the codes were made longer to avoid logic errors in Matlab).

The following step is to characterize $DT_{1,4}$. Based on the implementation of $F_{SA1,2}$, the vector characterizing $DT_{1,4}$ in terms of the data source, the data category, and the expected outputs is shown in Figure 5.14, with the following characteristics: $DS_x = DS_3$ = load sensor, $DC_x = DC_1$ = big data, O_1 = plots, and O_2 = hierarchical tree.

The third step concerns the execution of the sub-function $F_{SA1,3}$. It consists of matching the DST with DATs. As shown in Figure 5.15, the outcome of the execution of this sub-function is a vector of data analytics tools matching $DT_{1,4}$. These tools are T_2 = Decision trees, T_3 = Classification trees, T_5 = *K*-means, and T_6 = *K*-medoids. The outcome of the execution of sub-function $F_{SA1,4}$ is shown in Figure 5.16. The concrete outputs are the matrix of the sum of weights of tools [18, 13, 17, 12] and the ordering of the sum matrix [18, 17, 13, 12]. The eventual ordering of the appropriate tools is [T_6, T_5, T_2, T_3]. Finally, the outcome of the execution of sub-function $F_{SA1,5}$ is shown in Figure 5.17. Two outputs are expected and obtained, the

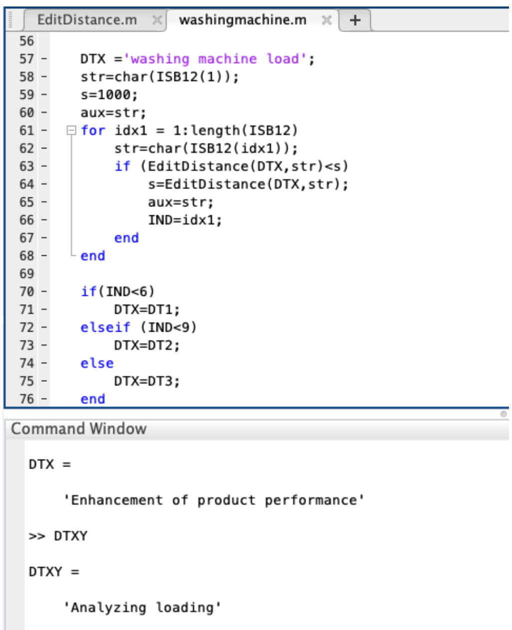


Figure 5.13. Process and outputs of the sub-function $F_{SA1,1}$

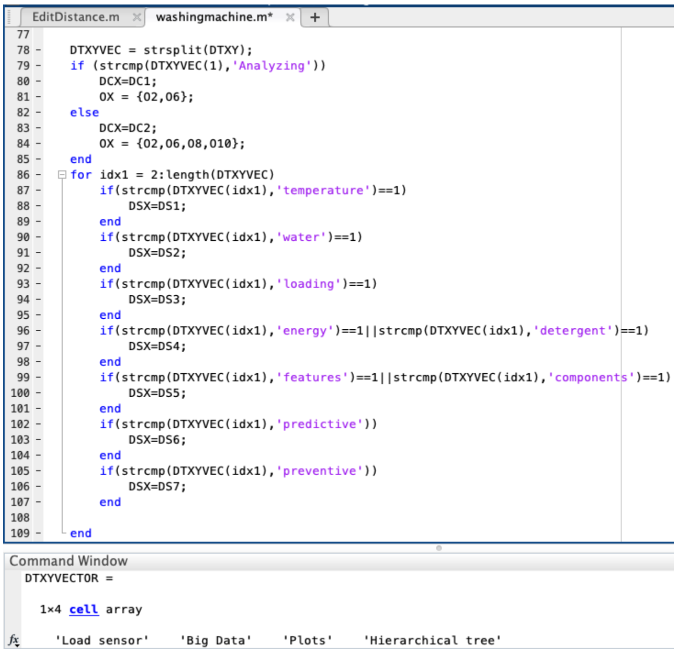


Figure 5.14. Process and outputs of the sub-function $F_{SA1,2}$

maximum sum of weights (18) and the corresponding tools to offer to the designer [T_6]. By referring back to Figure 5.12, we see the tools selected based on DTX (DT_1) are [T2, …, T6], and the recommended tool based on weight sum is [T6] (by calculating the weights sum). This means that F_{SA1} provides the best match.

The results of the testing are shown from Figure 5.13 through Figure 5.17. By analyzing these figures, the following points can be identified: (i) the algorithms can be implemented; (ii) they are computationally correct; (iii) the codes do not contain any errors; (iv) each algorithm set of inputs and outputs is identical to the expected results presented in the algorithm's description in Section 5.4.2; (v) the algorithms communicate with each other, since the output of one algorithm is the input for the following one; and (vi) no conversions of inputs or outputs were needed throughout the function implementation. We conclude that the flow of algorithms is coherent and that the recommendation function is a feasible and functional and can be implemented to serve its purpose.

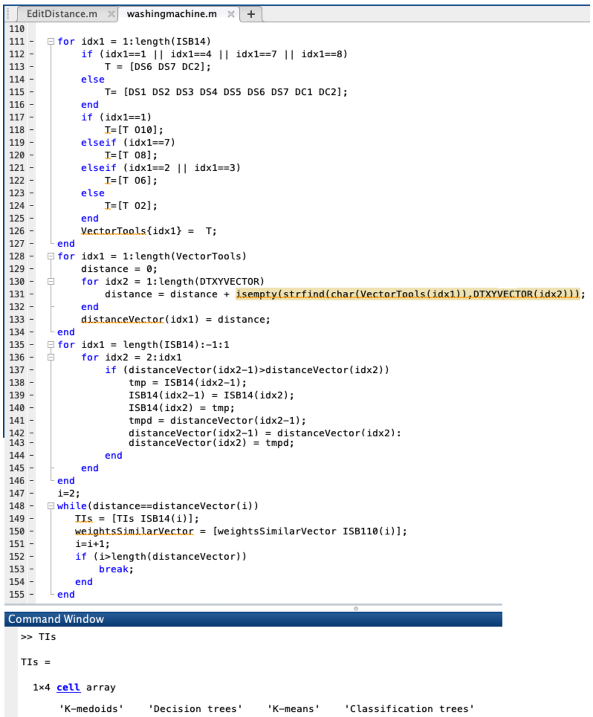


Figure 5.15. Process and outputs of sub-function $F_{SA1,3}$

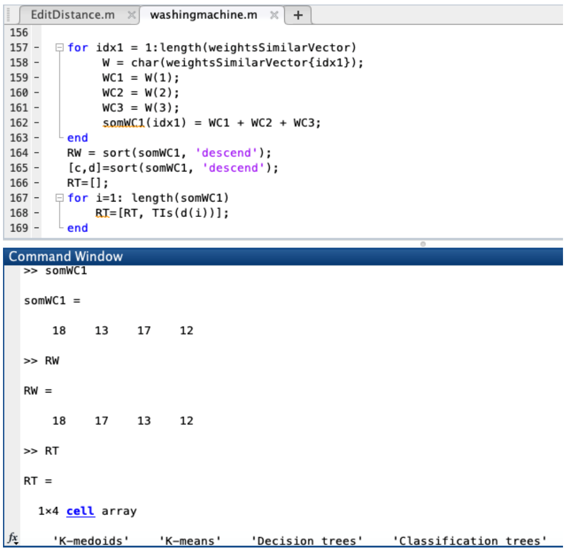


Figure 5.16. Process and outputs of sub-function $F_{SA1,4}$

On the other hand, it was observed that starting from the initial input DTX given by the designer, the final output was a finite matrix of DATs with high weight values. The function provided the same results computationally and manually. It was also demonstrated that the designer provided only the task, and the tools to use were automatically recommended. This leads to the conclusion that the function achieved its desired outputs. We can conclude that the task-relevant data analytics toolbox function (i) facilitates the choice process of designers, (ii) saves time and effort related to this matter, and (iii) covers the lack of designer's knowledge regarding DATs. The next generation SDATB in which this function will be implemented will be optimized for product designers. Instead of getting lost in the huge number of DATs and their new updates, which could take hours, designers will get their suitable tools in seconds. This will allow them to keep their focus on their kernel job of product enhancements rather than becoming distracted with an auxiliary function of investigating, studying, and comparing DATs that might or might not be suitable for their DTX.

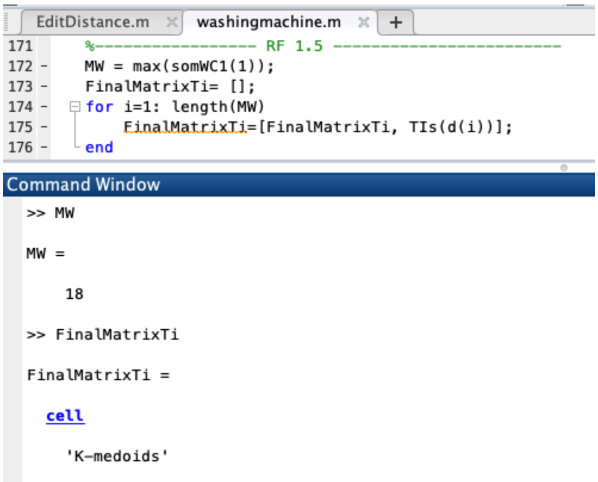


Figure 5.17. Process and outputs of sub-function $F_{SA1,5}$

5.5.3. Testing the functionality of the smart user identification module

The basic requirement for the smart user identification module is to offer a secure identification process and environment for the designer. To test it, we built a smart user identification module, including its GUI, in Matlab. We implemented and integrated all of the algorithms presented in Section 5.4.3. The developed simplified GUI and the first menu item the designer can access upon opening the SDATB are shown in Figure 5.18. The starting menu includes (i) "ADD NEW IMAGE," where the designer gives access to the computer's camera; (ii) "DATABASE," where the designer's database and attendance form are saved (see Figure 5.19); (iii) "RESET," which has almost the same

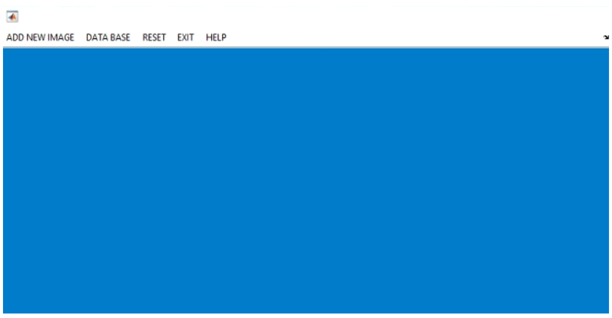


Figure 5.18. The starting menu of the smart data analytics toolbox

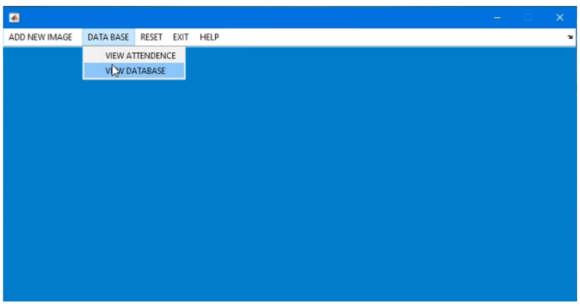


Figure 5.19. Options of the Database menu in the graphical user interface

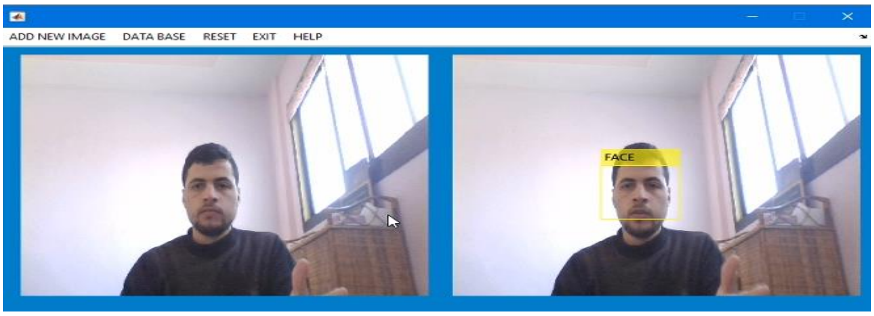


Figure 5.20. Face captured from live video

options as "DATABASE" and gives the user the choice to delete previous database and attendance records; (iv) "EXIT," which was added as an option to allow the designer to leave the SDATB at any time; and finally (v) a "HELP" option for users who do not know how to use the face identification function to log in to the toolbox.

Once the user choses to add a new image from the camera, the camera is connected to the SDATB, and the toolbox internally preprocesses the image and captures the face from the live video, as represented in Figure 5.20 (live video stream is shown on the left, and the captured face image is shown on the right). The yellow rectangle represents the detected captured face.

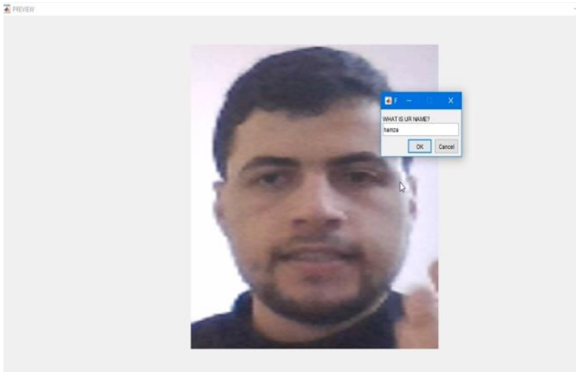


Figure 5.21. Name insertion for a new user of the smart data analytics toolbox

The following step is to recognize the captured face image. In the case of a new user who is still not registered in the database, the toolbox will ask the designer to enter his or her name, as shown in Figure 5.21. After saving the identity of the designer, the database stores one image (in .jpg format), which the designer can access at any time to check whether someone else has tried to use his or her identity to log in or the image has been deleted from the database. Figure 5.22 shows the user's database after the first identification.

To test the performance of the implemented algorithms, we asked the same user to log out of the SDATB and then log in again. This was done to test the face recognition algorithm and its capacity to recognize the user and recall the user's name. The face was detected and the training started to match it with the existing image of the user in the database. In a second, the SDATB was able to recognize the user, representing for him the image of his face captured at his last toolbox login and his name to justify the matching, as presented in Figure 5.23. On the left in this figure, the "current image" is represented, and on the right it is the "database entered image." The SDATB also informs the user that his attendance has been recorded, in case he would like to access his history of

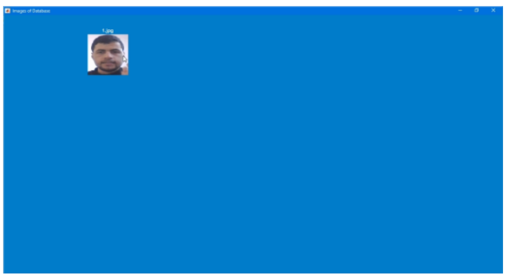


Figure 5.22. Database after the first identification of the designer

attendance. This is to insure extra security in toolbox usage. In addition to accessing the face images saved in the database, the SDATB keeps a record of the time and date of identification, as shown in Figure 5.24 related to the same user.

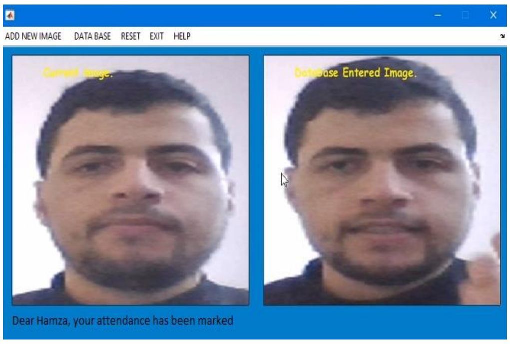


Figure 5.23. The result of face recognition

The obtained results imply that the smart user identification function did not present any computational errors, and the algorithms implemented were sufficient to achieve the needed objectives. The functionality testing validated that the proposed computational mechanism supports a secure identification and access to the SDATB by designers. It provides the opportunity to check whether the designer was the only one to access the toolbox. By referring to the face detection phase in Figure 5.20, we can see that the module was able to find the designer’s face in the image and crop around it. As for the face recognition, the toolbox was able to identify the user and provide him the needed information about his SDATB access. These elements allowed us to test both the feasibility and the performance of the implemented smart user identification module.

Name	Date	Time	Attendance
Hamza	04-Jan-2020	16:29:43	Present
Hamza	04-Jan-2020	16:35:31	Present

Figure 5.24. Attendance document (form) of the same user

5.6. Recognized limitations

Based on the research activities and the testing of the implemented modules, we recognized some limitations:

- ***Concerning the merging middle-of-life data streams recommendation module:***
  - The lack of published literature dedicated to understanding semantic inference in the context of product enhancement made it difficult to choose the optimal algorithms and techniques to deploy for our functionality. We chose some of the techniques used for semantic inference related to images, videos, and texts and converted them to the context of sensor data. This made our solution dependent on data stream annotations, since it requires past anomalies to be determined as well as textual possible recommendations.
  - The current statistical model suffers from the limitation of not being recurrent, and it simplifies decision-making only within a fixed-length time window.
  - Incorporating prior knowledge of product anomalies directed the focus of our implemented function towards maintenance types of action plans. We used this

to show how the semantics from multiple sensors can be captured and converted into an action plan. Developing algorithms able to automatically generate rules and to be aware of the dynamic changes in context and data streams (different from a washing machine as a product) will reduce the time and effort allocated to scenario building and algorithm training.

- Although using simulated MoLD-Ss allowed us to achieve the objective of testing the pilot implementation's functionality, the simulated MoLD-Ss could not completely substitute for real-life MoLD-Ss, since it did not consider the real-life behavior of sensors in the use environment. Therefore, we did not consider the performance and robustness of the computation in the functionality testing of the pilot implementation. In other words, this pilot implementation was not challenged by the presence of unexpected patterns in the data. In the case of real data streams, the processing may take longer time or may show inconsistencies.
- The implemented synthetic database does not assume multidimensional values for single-sensor streams. To improve the performance of the function, a model capable of reasoning on multidimensional time series is needed; however, this requires the adoption of a more complex neural network.
- We used Matlab's deep learning toolbox for the implementation of the function. However, such implementation was time consuming (in comparison to better computational solutions), since everything had to be written manually and from scratch (estimation, optimization, training, etc.) in Matlab. At the same time, other programming packages (Python, for example) feature predefined operations and procedures that are ready to use or adjust. While Matlab was a good choice to achieve our objective, it has its own limitations.

- ***Concerning the task-relevant data analytics tools recommendation module:***
  - Using machine learning algorithms might have reduced the time needed for code building, since some inputs (e.g. weights and tools) could be generated and adjusted automatically. This should be considered in future improvements of this recommendation function.
  - DTX are supposed to be known for the system. Syntactic matching is being implemented. This is one of the function limitations. Semantic matching based on synonyms is being considered for the future.
  - For the testing, a small range of system inputs were considered (DTX, DATs, etc.), which made the computing easy to realize. In the case of a wide range of inputs, the computing might be time consuming, and human error is likely. In the future, automation methods for inputs insertion are to be studied.
- ***Concerning the smart user identification module:***
  - The smart user identification module is implemented using face detection and recognition. It allows a safe environment but is not 100% safe. For example: the case of tweens. A possible enhancement would be the augmentation of this

function with identification of biometric features, such as fingerprints, which are unique for every person.

- The smart user identification module could perform face detection only if the face was almost motionless in front of the camera. This is a limitation. A more sophisticated mechanism needs to be implemented that allows face detection in a dynamic environment and even if the user is moving.
- The implemented module may not be able to recognize users wearing facial accessories such as glasses and may fail to register them as new users. This means that a better feature extraction algorithm needs to be considered for the future.

5.7. Conclusions

The implementation and functionality testing of the modules and algorithms of the demonstrative SDATB provided evidence confirming their correctness from logical and computational points of view. The following conclusions were deduced from the testing results:

- ***Concerning the merging middle-of-life data streams recommendation module:***
  - The proposed function is a data-driven one, capable of capturing semantics. It can be seen as a knowledge construction with the help of behavior encoder.
  - The function for merging MoLD-Ss is useful for helping designers understand unsupervised data and for assessing large volumes of sensor information.
  - The function is capable of processing vast amounts of data streams to discover unusual behavior in MoLD-Ss.
  - The implementation of such a function will shorten the time for decision-making for product maintenance, repair, and enhancement. It not only lists the anomalies related to products but also recommends an action plan with the next steps to be taken to adjust the product.
  - The implemented function is capable of deriving simple yet efficient knowledge representation with the help of a triplet network.
  - Semantic reasoning is implemented as a similarity search function. It allows measurement of the similarity of two behavior patterns as a distance in the latent space. To capture the meaning conveyed by the individual situations, we applied a novel mechanism of attention that learns the most important data streams for each situation without requiring the annotation of these dependencies in the knowledge base.
  - By merging MoLD-Ss, more information can be provided than by each sensor individually. Therefore, the implementation of the proposed recommendation function has practical benefits. It makes the actual state and condition of the product transparent and communicates that state to the designer while the product is in use by its owner.

- The realized function offers a recommendation to designer that is semantically correct based on product anomalies. It reduces the time and effort of processing data streams and makes decision-making on enhancements a fast process.

- ***Concerning the task-relevant data analytics tools recommendation module:***
  - The recommendation function is able to interpret designers' inputs and propose a description of the design task specified by the designer (DTX).
  - The recommendation function reasons with DATs and recommends the one that best matches DTX.
  - The rule for DAT selection and recommendation is captured by their weights and the matrix that matches DTs and DATs.
  - The recommended tool was proven to be the most appropriate one based on DTX.
  - The recommendation function compensates for product designers' lack of knowledge about the use of DATs in particular tasks.
  - The recommendation function reduces the time and effort associated with tool selection.
- ***Concerning the smart user identification module:***
  - The proposed approach to user identification is able to recognize faces based on images made even by low-quality, low-resolution cameras. This is shown in the figures presented in Section 5.5.3.
  - The combination of algorithms improves face detection and recognition of users in various poses and in front of difficult backgrounds. This reduces the rate of false negative cases and detects faces in images having different imaging resolution and taken under different lighting conditions.
  - The implemented smart user identification function allows the user to access the toolbox just by using a laptop camera. This form of authentication is more efficient and less time consuming than using security information such as passwords.
  - The multiplicity of algorithms discussed in the literature allowed us to choose the most efficient algorithm and data constructs to offer an up-to-date smart user identification mechanism.

5.8. References

[1]. Veeramanikandan, K., Ezhilarasi, R., & Brindha, R. (2013). An FPGA based real time face detection and recognition system across illumination. *International Journal of Engineering Science and Engineering*, 1(5), 66-68.

[2]. Nikisins, O., Fuksis, R., Kadikis, A., & Greitans, M. (2015). Face recognition system on Raspberry Pi. In: *Proceedings of the International Conference on Information Processing and Control Engineering*, x-x.

[3]. Gravina, R., Alinia, P., Ghasemzadeh, H., & Fortino, G. (2017). Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35, 68-80.

[4]. Seiti, H., & Hafezalkotob, A. (2018). Developing pessimistic–optimistic risk-based methods for multi-sensor fusion: An interval-valued evidence theory approach. *Applied Soft Computing*, 72, 609-623.

[5]. Xiao, F. (2019). Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. *Information Fusion*, 46, 23-32.

[6]. Shen, L., Tay, F.E., Qu, L., & Shen, Y. (2000). Fault diagnosis using rough sets theory. *Computers in industry*, 43(1), 61-72.

[7]. Jiang, W., & Hu, W. (2018). An improved soft likelihood function for Dempster–Shafer belief structures. *International Journal of Intelligent Systems*, 33(6), 1264-1282.

[8]. Jiang, W., & Wei, B. (2018). Intuitionistic fuzzy evidential power aggregation operator and its application in multiple criteria decision-making. *International Journal of Systems Science*, 49(3), 582-594.

[9]. Fu, C., Yang, J.B., & Yang, S.L. (2015). A group evidential reasoning approach based on expert reliability. *European Journal of Operational Research*, 246(3), 886-893.

[10]. Mello, R.F., & Ponti, M.A. (2018). Machine learning: a practical approach on the statistical learning theory. Springer.

[11]. Kang, B., Chhipi-Shrestha, G., Deng, Y., Hewage, K., & Sadiq, R. (2018). Stable strategies analysis based on the utility of Z-number in the evolutionary games. *Applied Mathematics and Computation*, 324, 202-217.

[12]. Zhou, X., Deng, X., Deng, Y., & Mahadevan, S. (2017). Dependence assessment in human reliability analysis based on D numbers and AHP. *Nuclear Engineering and Design*, 313, 243-252.

[13]. Xu, S., Jiang, W., Deng, X., & Shou, Y. (2018). A modified Physarum-inspired model for the user equilibrium traffic assignment problem. *Applied Mathematical Modelling*, 55, 340-353.

[14]. Zhang, Q., Li, M., & Deng, Y. (2018). Measure the structure similarity of nodes in complex networks based on relative entropy. *Physica A: Statistical Mechanics and its Applications*, 491, 749-763.

[15]. Zhou, X., Hu, Y., Deng, Y., Chan, F.T., & Ishizaka, A. (2018). A DEMATEL-based completion method for incomplete pairwise comparison matrix in AHP. *Annals of Operations Research*, 271(2), 1045-1066.

[16]. Fu, C., Xu, D.L., & Yang, S.L. (2016). Distributed preference relations for multiple attribute decision analysis. *Journal of the Operational Research Society*, 67(3), 457-473.

[17]. Pawlak, Z. (2012). Rough sets: Theoretical aspects of reasoning about data. Springer Science & Business Media.

[18]. Zhou, B., Cho, H., & Mansfield, A. (2017). Robust Sensor Data Fusion Through Adaptive Threshold Learning. In: *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, Cham, 277-282.

[19]. Deng, X., Han, D., Dezert, J., Deng, Y., & Shyr, Y. (2015). Evidence combination from an evolutionary game theory perspective. *IEEE Transactions on Cybernetics*, 46(9), 2070-2082.

[20]. Zadeh, L.A. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2), 85-85.

[21]. Mardani, A., Jusoh, A., & Zavadskas, E.K. (2015). Fuzzy multiple criteria decision-making techniques and applications–Two decades review from 1994 to 2014. *Expert systems with Applications*, 42(8), 4126-4148.

[22]. Rahaman, S., & Hossain, M.S. (2013). A belief rule based clinical decision support system to assess suspicion of heart failure from signs, symptoms and risk factors. In: *Proceedings of the International Conference on Informatics, Electronics and Vision*, IEEE, 1-6.

[23]. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. MIT press.

[24]. Cappello, F., Sabatini, R., & Ramasamy, S. (2015). Multi-sensor data fusion techniques for RPAS detect, track and avoid. In: *Proceedings of the SAE 2015 AeroTech Congress and Exhibition*, Warrendale, PA, United States, 1-12.

[25]. Kumar, T. P., Saimurugan, M., Haran, R.H., Siddharth, S., & Ramachandran, K.I. (2019). A multi-sensor information fusion for fault diagnosis of a gearbox utilizing discrete wavelet features. *Measurement Science and Technology*, 30(8), 1-11.

[26]. Zhou, Q., Chen, G., Jiang, W., Li, K., & Li, K. (2019). Automatically detecting excavator anomalies based on machine learning. *Symmetry*, 11(8), 957.

[27]. Nweke, H.F., Teh, Y.W., Mujtaba, G., Alo, U.R., & Al-garadi, M.A. (2019). Multi-sensor fusion based on multiple classifier systems for human activity identification. *Human-centric Computing and Information Sciences*, 9(1), 34.

[28]. Basurto, N., & Herrero, Á. (2019). Data selection to improve anomaly detection in a component-based robot. In: *Proceedings of the International Workshop on Soft Computing Models in Industrial and Environmental Applications*, Springer, Cham, 241-250.

[29]. Luo, H., Jia, S., & Zhang, W. (2019). Hierarchical temporal memory based anomaly detection for hydrological monitoring of unmanned surface vehicle. In: *Proceedings of the 2nd International Conference on Information Communication and Signal Processing*, IEEE, 420-424.

[30]. Von Luxburg, U., & Schölkopf, B. (2011). Statistical learning theory: Models, concepts, and results. *Handbook of the History of Logic*, 10, 651-706.

[31]. Utkin, L.V., Zhuk, Y.A., & Chekh, A.I. (2013). An ensemble-based feature selection algorithm using combination of support vector machine and filter methods for solving classification problems. *European Journal of Technology and Design*, (1), 70-76.

[32]. Utkin, L.V., & Coolen, F.P. (2014). Classification with support vector machines and Kolmogorov-Smirnov bounds. *Journal of Statistical Theory and Practice,* 8(2), 297-318.

[33]. Anthony, M.H.G., & Biggs, N. (1997). Computational learning theory, 30. Cambridge University Press.

[34]. Noureldin, A., Sharaf, R., Osman, A., & El-Sheimy, N. (2004). INS/GPS data fusion technique utilizing radial basis functions neural networks. In: *Proceedings of the Position Location and Navigation Symposium*, IEEE, 280-284.

[35]. Palm, G., & Schwenker, F. (2009). Sensor-fusion in neural networks. *Harbour Protection Through Data Fusion Technologies*, Springer, Dordrecht, 299-306.

[36]. Shahbazian, E., Rogova, G., & de Weert, M.J. (2008). Harbour protection through data fusion technologies. Springer Science & Business Media.

[37]. Scherr, T., Bartschat, A., Reischl, M., Stegmaier, J., & Mikut, R. (2018). Best practices in deep learning-based segmentation of microscopy images. In: *Proceedings of the 28th Workshop Computational Intelligence, Dortmund,* KIT Scientific Publishing, 2018175.

[38]. Liu, Q. (2017). Word sense determination from Wikipedia data using neural networks. Doctoral dissertation, San Jose State University.

[39]. Becker, M., Lippel, J., Stuhlsatz, A., & Zielke, T. (2020). Robust dimensionality reduction for data visualization with deep neural networks. Graphical Models, 108, 101060.

[40]. Xiao, Y., Wu, J., Lin, Z., & Zhao, X. (2018). A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using RNA-seq data. *Computer Methods and Programs in Biomedicine*, 166, 99-105.

[41]. Paradarami, T.K., Bastian, N.D., & Wightman, J.L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, 83, 300-313.

[42]. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1), 1-38.

[43]. Paradarami, T.K., Bastian, N.D., & Wightman, J.L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, 83, 300-313.

[44]. Artiba, A., Riane, D.F., Nourelfath, M., & Nahas, N. (2005). Artificial neural networks for reliability maximization under budget and weight constraints. *Journal of Quality in Maintenance Engineering*, 11(2), 139-151.

[45]. Bathla, G., Aggarwal, H., & Rani, R. (2019). Using deep learning to improve recommendation with direct and indirect social trust. *Journal of Statistics and Management Systems*, 22(4), 665-677.

[46]. Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[47]. Häusser, P. (2018). Learning by association. Doctoral dissertation, Technische Universität München.

[48]. Wang, Q. (2018). Exponential convergence of the deep neural network approximation for analytic functions. *arXiv preprint arXiv:1807.00297*.

[49]. Arora, S., Cohen, N., Golowich, N., & Hu, W. (2018). A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*.

[50]. Shamir, O. (2018). Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. *arXiv preprint arXiv:1809.08587*.

[51]. Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

[52]. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[53]. Larriba, Y., Rueda, C., Fernández, M.A., & Peddada, S.D. (2019). Microarray data normalization and robust detection of rhythmic features. *Microarray Bioinformatics*, Humana, New York, 207-225.

[54]. Shen, X., Gong, X., Cai, Y., Guo, Y., Tu, J., Li, H., ... & Zhu, Z. J. (2016). Normalization and integration of large-scale metabolomics data using support vector regression. *Metabolomics*, 12(5), 89-101.

[55]. Oord, A.V.D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[56]. Mohammed, E.A., Naugler, C.T., & Far, B.H. (2018). A Sliding-Window algorithm implementation in MapReduce. *Applications of Data Management and Analysis*, Springer, Cham, 69-83.

[57]. Yang, X., Ramesh, P., Chitta, R., Madhvanath, S., Bernal, E.A., & Luo, J. (2017). Deep multimodal representation learning from temporal data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* IEEE, 5447-5455.

[58]. Ray, A., Kumar, S., Reddy, R., Mukherjee, P., & Garg, R. (2019). Multi-level attention network using text, audio and video for depression prediction. In: *Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop,* 81-88.

[59]. Chen, C.C., Huang, H.H., & Chen, H.H. (2019). Next cashtag prediction on social trading platforms with auxiliary tasks. In: *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, 525-527.

[60]. Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. In: *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*, Springer, Cham, 84-92.

[61]. Shah, D., Shah, P., & Banerjee, A. (2017). Similarity based regularization for online matrix-factorization problem: An application to course recommender systems. In: *Proceedings of TENCON Region 10 Conference*, IEEE, 1874-1879.

[62]. Zhang, F., Gong, T., Lee, V.E., Zhao, G., Rong, C., & Qu, G. (2016). Fast algorithms to evaluate collaborative filtering recommender systems. *Knowledge-Based Systems*, 96, 96-103.

[63]. Cheruvu, A., Rashakrishna, V., & Rajasekhar, N. (2017). Using normal distribution to retrieve temporal associations by Euclidean distance. In: *Proceedings of the International Conference on Engineering & MIS*, IEEE, 1-3.

[64]. Jaafor, O., & Birregah, B. (2017). Sponge walker: community detection in large directed social networks using local structures and random walks. In: *Proceedings of the European Network Intelligence Conference*, Springer, Cham. 143-152.

[65]. Yu, M., Wang, J., Li, G., Zhang, Y., Deng, D., & Feng, J. (2017). A unified framework for string similarity search with edit-distance constraint. *The International Journal on Very Large Data Bases*, 26(2), 249-274.

[66]. Sondur, M.S.D., Chigadani, M.A.P., & Nayak, S. (2016). Similarity measures for recommender systems: A comparative study. *Journal of Research*, 2(3), 76-80.

[67]. Magara, M.B., Ojo, S.O., & Zuva, T. (2018). A comparative analysis of text similarity measures and algorithms in research paper recommender systems. In: *Proceedings of the Conference on Information Communications Technology and Society*, IEEE, 1-5.

[68]. Patil, H., & Thakur, R.S. (2018). Document clustering: A summarized survey. *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications*, IGI Global, 47-64.

[69]. Zhao, Q., Bhowmick, S.S., & Gruenwald, L. (2006). Cleopatra: Evolutionary pattern-based clustering of web usage data. In: *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, 323-333.

[70]. Gan, Y., Guan, J., & Zhou, S. (2009). A pattern-based nearest neighbor search approach for promoter prediction using DNA structural profiles. *Bioinformatics*, 25(16), 2006-2012.

[71]. Blake, M.B., & Nowlan, M.F. (2007). A web service recommender system using enhanced syntactical matching. In: *Proceedings of the International Conference on Web Services*, IEEE, 575-582.

[72]. Li, C., Wang, B., & Yang, X. (2007). VGRAM: Improving performance of approximate queries on string collections using variable-length grams. *The International Journal on Very Large Data Bases*, 7, 303-314.

[73]. Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), 707-710.

[74]. Lam, L. (2008). An introduction to R. Business and Decision Amsterdam.

[75]. Beak, A. (2017). Strings and patterns. *PHP 7 Zend Certification Study Guide*, Apress, Berkeley, CA, 57-81.

[76]. Budd, T. (1994). Data Structures in C+. Addison-Wesley.

[77]. Ganesh, E.N. (2017). Steganalysis of LSB Insertion Method in Uncommpressed Images Using Matlab. *International Journal of Digital Communication and Analog Signals*, 3(1), 1-6.

[78]. Appiah, O., & Martey, E.M. (2015). Magnetic Bubble Sort Algorithm. *International Journal of Computer Applications*, 122(21), 24-28.

[79]. Uehara, R. (2019). Algorithms for Searching and Sorting. *First Course in Algorithms Through Puzzles, Springer, Singapore*, 57-87.

[80]. Hanna, K.J. (2018). *U.S. Patent No. 9,959,478*. Washington, DC: U.S. Patent and Trademark Office.

[81]. Dharani, P., & Vibhute, A.S. (2017). Face recognition using wavelet neural network. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(5), 101-107.

[82]. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., ... & Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 5265-5274.

[83]. Sikkandar, H., & Thiyagarajan, R. (2019). Soft biometrics-based face image retrieval using improved grey wolf optimisation. *IET Image Processing*, 14(3), 451-461.

[84]. Ahamed, H., Alam, I., & Islam, M.M. (2018). HOG-CNN based real time face recognition. In: *Proceedings of the International Conference on Advancement in Electrical and Electronic Engineering*, IEEE, 1-4.

[85]. Zhao, W., Chellappa, R., Phillips, P.J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4), 399-458.

[86]. Lahiani, H., Kherallah, M., & Neji, M. (2016). Hand pose estimation system based on Viola-Jones algorithm for android devices. In: *Proceedings of the 13th International Conference of Computer Systems and Applications*, IEEE, 1-6.

[87]. Lescano, G. E., Santana Mansilla, P., & Costaguta, R. (2017). Analysis of a GPU implementation of Viola-Jones' algorithm for features selection. *Journal of Computer Science & Technology*, 17(1), 68-73.

[88]. Laddha, P., & Padalkar, G.R. (2018). Automatic Facial Expression Recognition System. *International Journal of Pure and Applied Mathematics*, 118(24), 1-10.

[89]. Alionte, E., & Lazar, C. (2015). A practical implementation of face detection by using Matlab cascade object detector. In: *Proceedings of the 19th International Conference on System Theory, Control and Computing*, IEEE, 785-790.

[90]. George, G., Boben, R., Radhakrishnan, B., & Suresh, L.P. (2017). Face recognition on surgically altered faces using principal component analysis. In: *Proceedings of the International Conference on Circuit, Power and Computing* Technologies, IEEE, 1-6.

[91]. Jain, U., Choudhary, K., Gupta, S., & Privadarsini, M.J.P. (2018). Analysis of face detection and recognition algorithms using viola jones algorithm with PCA and LDA. In: *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics*, IEEE, 945-950.

[92]. Hefenbrock, D., Oberg, J., Thanh, N.T.N., Kastner, R., & Baden, S. B. (2010). Accelerating Viola-Jones face detection to FPGA-level using GPUs. In: *Proceedings of the 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, IEEE, 11-18.

[93]. Song, H., & Gao, Y. (2017). Preceding vehicle distance computation based on dark prior. In: *Proceedings of the 36th Chinese Control Conference*, IEEE, 9952-9959.

[94]. Jindal, R., Gupta, A., & Vatta, S. (2013). Face detection using digital image processing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(11).

[95]. Haseeb, M. S., Ayub, R.M.B., Ali, M.N., & Khan, M.A. (2017). Face and face parts detection in image processing. *Lahore Garrison University Research Journal of Computer Science and Information Technology*, 1(1), 62-68.

[96]. Ananthakumar, A. (2018). efficient face and gesture recognition for time sensitive application. In: *Proceedings of the Southwest Symposium on Image Analysis and Interpretation*, IEEE, 117-120.

[97]. Viola, P., & Jones, M.J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 137-154.

[98]. Nazeer, S. A., Omar, N., Jumari, K.F., & Khalid, M. (2007). Face detecting using artificial neural network approach. In: *Proceedings of the 1st Asia International Conference on Modelling & Simulation*, IEEE, 394-399.

[99]. Aït-Sahalia, Y., & Xiu, D. (2019). Principal component analysis of high-frequency data. *Journal of the American Statistical Association*, 114(525), 287-303.

[100]. Bartlett, M. S., Movellan, J.R., & Sejnowski, T.J. (2002). Face recognition by independent component analysis. *IEEE Transactions On Neural Networks*, 13(6), 1450-1464.

[101]. Li, Y., Gong, S., Sherrah, J., & Liddell, H. (2004). Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5), 413-427.

[102]. Patil, V.D. (2018). A review on face recognition techniques. In: *Proceedings of the 8th Nation Conference on Emerging trends in Engineering and Technology*, 7(1), 517-523.

[103]. Nefian, A.V., & Hayes, M.H. (1998). Hidden Markov models for face recognition. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, IEEE, 5, 2721-2724.

[104]. Gupta, P., Saxena, N., Sharma, M., & Tripathi, J. (2018). Deep neural network for human face recognition. *International Journal of Engineering and Manufacturing*, 8(1), 63-71.

[105]. Khan, S.M.H., & Hussain, A. (2016). Content based image retrieval using uniform local binary patterns. In: Proceedings of the International Conference on Data Mining, Multimedia, Image Processing and their Applications, 49-62.

[106]. Gabotshajwe, E. (2019). A critical evaluation of current published face recognition systems research aimed at improving security for atm transactions. *Selected Computing Research Papers*, 7-11.

[107]. Alomar, F.A., Muhammad, G., Aboalsamh, H., Hussain, M., Mirza, A.M., & Bebis, G. (2013). Gender recognition from faces using bandlet and local binary patterns. In: *Proceedings of the 20th International Conference on Systems, Signals and Image Processing*, IEEE, 59-62.

[108]. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., & Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6), 703-715.

[109]. Park, S.J., Park, D.K., & Won, C.S. (2000). Core experiments on MPEG-7 edge histogram descriptor. *MPEG document M5984*, Geneva.

[110]. Jyothi, B., Madhavee Latha, Y., & Mohan, P.K. (2015). An effective multiple visual features for content based medical image retrieval. In: *Proceedings of the 9th International Conference on Intelligent Systems and Control*, IEEE, 1-5.

[111]. Rahman, S., Banik, P., & Naha, S. (2014). LDA based paper currency recognition system using edge histogram descriptor. In: *Proceedings of the 17th International Conference on Computer and Information Technology*, IEEE, 326-331.

[112]. Rahman, S., Naim, S.M., Farooq, A.A., & Islam, M. (2011). Performance of PCA based semi-supervised learning in face recognition using MPEG-7 edge histogram descriptor. *Journal of Multimedia*, 6(5), 404-415.

[113]. Mehta, P., & Tomar, P. (2016). An efficient attendance management system based on face recognition using Matlab and Raspberry Pi 2. *International Journal of Engineering Technology Science and Research*, 3(5), 71-78.

[114]. Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6), 765-781.

[115]. Kapil, D., & Jain, A. (2013). A brief review on feature based approaches for face recognition. *International Journal of Science and Research*, 4(5), 273-277.

[116]. Bennamoun, M., Guo, Y., & Sohel, F. (1999). Feature selection for 2D and 3D face recognition. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1-28.

[117]. Annesley, J., Orwell, J., & Renno, J.P. (2005). Evaluation of MPEG7 color descriptors for visual surveillance retrieval. In: *Proceedings of the International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, IEEE, 105-112.

[118]. Kim, J.H., Lim, H.Y., & Kang, D.S. (2008). An implementation of the video retrieval system by video segentation. In: *Proceedings of the 14th Asia-Pacific Conference on Communications*, IEEE, 1-5.

[119]. Plichoski, G.F., Dorini, L., Lopes, H.S., & Chidambaram, C. (2016). A cascade face recognition approach with illumination compensation using local features. In: *Proceedings of the 12th Workshop de Visao Computacional*, 340-345.

[120]. Kurchaniya, D., & Johari, P.K. (2017). An efficient approach for image retrieval using particle swarm optimization. *International Journal of Computer Sciences and Engineering*, 5(6), 90-99.

[121]. Kusumaningrum, R., & Arymurthy, A.M. (2011). Color and texture feature for remote sensing-image retrieval system: a comparative study. *International Journal of Computer Science Issues*, 8(5), 125-135.

[122]. Khokher, A., & Talwar, R. (2017). A fast and effective image retrieval scheme using color-, texture-, and shape-based histograms. *Multimedia Tools and Applications*, 76(20), 21787-21809.

[123]. Song, C., & Zhang, Q. (2008). Sliding-window algorithm for asynchronous cooperative sensing in wireless cognitive networks. In: *Proceedings of the International Conference on Communications*, IEEE, 3432-3436.

[124]. Weinberger, K.Q., Blitzer, J., & Saul, L.K. (2006). Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 1473-1480.

[125]. Zou, Y., Dong, F., Lei, B., Sun, S., Jiang, T., & Chen, P. (2014). Maximum similarity thresholding. *Digital Signal Processing*, 28, 120-135.

[126]. Bao, F., & Maier, T. (2020). Stochastic gradient descent algorithm for stochastic optimization in solving analytic continuation problems. *Foundations of Data Science*, 2(1), 1-17.

[127]. Ribeiro, R., Lopes, D., & Neves, A. (2018). Access control in the wild using face verification. *Intelligent Video Surveillance*. IntechOpen, 1-18.

[128]. Batur, A.U., Flinchbaugh, B.E., & Hayes, M.H. (2003). A DSP-based approach for the implementation of face recognition algorithms. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, IEEE, 2, 253-256.

[129]. Faraji, M.R., & Qi, X. (2016). Face recognition under varying illuminations using logarithmic fractal dimension-based complete eight local directional patterns. *Neurocomputing*, 199, 16-30.

[130]. Parihar, V.R., & Dhote, A.P. (2017). A Novel Approach to Real Time Face Detection and Recognition. *International Journal of Computer Sciences and Engineering*, 5(9), 62-67.

[131]. Kumar, S., Singh, S., & Kumar, J. (2019). Multiple face detection using hybrid features with SVM classifier. *Data and Communication Networks*, Springer, Singapore, 253-265.

[132]. El Kaddouhi, S., Saaidi, A., & Abarkan, M. (2017). Eye detection based on the Viola-Jones method and corners points. *Multimedia Tools and Applications*, 76(21), 23077-23097.

[133]. Maia, D., & Trindade, R. (2016). Face detection and recognition in color images under MATLAB. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9(2), 13-24.

[134]. Knight, K.D. (2017). *U.S. Patent Application No. 14/796,*484.

Chapter 6

Overall conclusions, reflections and recommendations

6.1. Reflections on methodologies and results

The main objective of the Ph.D. research was to determine the fundamentals and functions of a next generation SDATB. The objective of this toolbox is to support white goods designers in enhancing their products based on MoLD. From this perspective, four research cycles have been completed in this Ph.D. research: (i) establishing a knowledge platform for data analytics technologies, (ii) deriving a qualitative theory and framework underpinning the development of a smart toolbox, (iii) conceptualizing demonstrative SDATB functions, and (iv) implementing and testing the functionality of the demonstrative SDATB. This section provides reflection on and self-evaluation of the work and findings based on the results of the completed research cycles. It exposes (i) results of knowledge aggregation, (ii) activities of ideation and conceptualization of the SDATB, (iii) integration of the findings related to the functions of a demonstrative concept of the SDATB, (iv) implementation of the chosen functions, and finally (v) reporting on the outcomes of the functionality validation of the functions of the demonstrative SDATB prototype in a particular application case.

6.1.1. Reflections on the knowledge aggregation for data analytics technologies

The objective of the first research cycle was to overview and critically investigate the state of the art of existing data analytics tools dedicated to supporting product and service enhancement based on MoLD. The knowledge aggregation explored four domains of interest: (i) the nature of data, (ii) data transformation steps and techniques, (iii) data transformation means (tools and packages), and (iv) applications of data analytics. These domains were analyzed to determine the current state of, the knowledge gaps in, and the limitations of existing data analytics tools and techniques in the context of product enhancement by product designers using MoLD.

- **Reflections on the methodology:**

To achieve a comprehensive literature study, we opted for both qualitative and quantitative studies. The qualitative study used a two-phase methodology: (i) shallow and (ii) deep explorations. The first phase used VOSviewer software to develop a graphical topographic landscape of publications related to the research phenomenon based on a wide range of keywords. It illustrated the distribution of clusters of keyword-related publications as well as their peaks and plains. This visual representation transparently exposed four domains of interest to be investigated in a transitive ordering (as presented in Chapter 2). The second phase explored multiple publication sources (web repositories, conference proceedings, journals, and so on), which helped not only in defining more specific keywords within clusters but also in providing a quantitative characterization of interrelationships between keywords belonging to the same cluster.

Using one of the graphical representations available through VOSviewer, we were able to distinguish the strength of relationships among clusters and keywords and their significance and relatedness. This representation shows the complexity of the study and also indicates the key terms not to be studied separately due to their interconnections reflected by the authors of the publications studied. The information obtained from both phases of the quantitative part of the literature study was used in developing the reasoning model of the qualitative study. This part consisted of interpreting the findings and disclosing semantic relationships. The methodology used in this research cycle helped provide a structured knowledge platform and insights on data analytics techniques and technologies but did not provide a deep understanding of the opportunities and limitations that could be directly used in developing new DATs for product designers.

- **Reflections on the results:**

The main findings of the knowledge aggregation in the first research cycle are the following:

> ***The emergence of diverse big data drives the need to upgrade current data processing tools and toolboxes.***

Data produced by products is big and diverse. These characteristics make it complex and not easily interpreted. The combination of engineering, technical, and social data augment the diversity in data and the production of both qualitative and quantitative data that have to be combined in processing. This means that descriptive, prescriptive, predictive, and operational data have to be managed to determine the value of data. These data aspects are unfortunately not tackled together by existing data analytics tools. To deal with this data engineering challenge, efforts must be dedicated to adapting and modifying data analytics technologies to be able to process hidden patterns in data and the mixed semantics caused by the mentioned diversity.

> ***Middle-of-life data offers important opportunities for product enhancement on a continuous basis, but it remains neglected in the design process.***

MoLD are collected when the product is in use by the customer. MoLD includes use, service, and maintenance data. These data are the richest in the product lifecycle, since they are diverse and report on how the product is used by different customers, but also because the MoL stage is much longer than the beginning and end of life stages. MoLD informs about the failure, performance, age, operating environment, usage intensity, maintenance, refund, and replacement of the product. The continuous production of MoLD allows them to be transformed into knowledge for perpetual and long-term design improvements and optimization of maintenance operations based on customers' experiences. Despite the richness and importance of MoLD, companies barely utilize them, since conventional information systems used in the definition of products and services cannot collect and analyze MoLD. These facts beg the need for dedicated data analytics approaches and tools able to handle MoLD.

Existing data analytics tools are not tailored to the needs of product designers.

Currently available data analytics tools require a certain level of skill and knowledge of how to write and use algorithms for analytics. This aspect makes them in most cases difficult for product designers to use. DATs are supposed to help designers make decisions about the best possible enhancement opportunities, but this help is not offered to designers by the available tools, which are not dedicated to the changes in design tasks and strategies. What also falls short of designers' expectations is the lack of semantic interpretation of the outputs offered by the tools.

6.1.2. Reflections on building theory and framework for next generation smart data analytics toolbox

The second research cycle aimed to define a set of requirements and fundamentals for a next generation data analytics toolbox. The first activity was a web-based interrogation of white goods designers to investigate their needs, opinions, and expectations related to data analytics technologies. The outcome of this first activity was a theory of needs explaining what designers miss and need regarding DATs. The second activity was the development of ATF and the application of it in developing a new theory for a next generation SDATB. This new theory listed requirements for an SDATB dedicated for product designers.

- **Reflections on the methodology:**

In this research cycle, two main activities where planned and executed: (i) inductive and (ii) deductive studies. The inductive study was a bottom-up knowledge aggregation. It used a web-hosted QBI for white goods practical designers. It helped identify their needs and provided insights on what they miss in existing software tools based on their daily design activities related to white goods as well as their experience using data analytics technologies for product improvement. The outcome was a theory of white goods designers' needs reflecting on what they want in new data analytics technologies. Nevertheless, some challenges were encountered in conducting this inductive study.

When the research project was started, our topic was in its infancy, which created certain technical difficulties. Namely, (i) both literature and data on the use of DATs in design contexts were limited, (ii) sampling participants for the interrogation was delicate, since they deal with various types of products, and (iii) the completion of the explanatory and meaningful questionnaire. If the study had been conducted this year, when more work has been dedicated to DATs' enhancement and contextualization, the literature study would have been richer, and the organization of the interrogation and the formulation of the needs would have been sharper.

On the other hand, the deductive approach was a bottom-up knowledge aggregation investigation of theories related to DATs development. It involved axiomatization-based conceptual discretization of relevant theories, incorporating semantic fusion of axioms and postulates extracted from the textual formulations of theories into the body of a new and synthetic explanatory theory. The outcome was the creation of a new theory consolidating requirements and fundamentals of a new SDATB that takes into account practical white goods designer's needs reflected by the need theory. The first step of this study consisted of building a new methodology of theorizing in a deductive manner, called ATF, and using it to develop a new theory based on five component theories.

The approach used provided sufficient knowledge on next generation data analytics tools and toolbox fundamentals and requirements. From a methodological point of view, ATF is novel but also complex. The literature investigation proved that no preliminary work has been published applying reasoning similar to that of ATF in merging design engineering theories. This reflects the novel and unique aspects of this methodology. The complexity was caused first by converting axiomatization mathematical principles into a design context and second by the manual work done in major parts of the approach to fully capture the semantics of the component theories. It was indeed beneficial but time consuming due to possible human error and revisions to avoid errors, as well as to the need for decomposing theories into axioms and postulates and for elaborating representations such as relationships networks. More than 150 hours were dedicated to ATF. In the future, efforts have to be dedicated to the automation of the methodology steps.

- **Reflections on the results:**

The major findings of both inductive and deductive activities are the following:

> ***There is a mismatch between the efforts and challenges documented in the literature and the expectations of designers of white goods regarding new data analytics computer support.***

From the literature study executed in the first research cycle, we concluded that most efforts were put into augmenting the technical performances of the tools and adapting them to the overwhelming variety and volume of big data without really considering the context or the tool users. The global assumption was about the lack of sharpness of the tools, and all efforts were oriented accordingly. In the case of white goods designers, however, the performances and the semantics formed just one component within their list of expectations. It was observed in the answers given by participants in the web-based interrogation that the interfacing and data management were also important aspects and needs. These designers want to be assisted and advised while using these tools. They want the tool to be smart, intuitive, customizable, affordable, and accessible anywhere and at any time. Consequently, developers of new data analytics tools need to consider the expectations of the specific users. A tool designed for data analysts is definitely not the tool wanted by white goods designers.

> ***Most of the needs formulated by designers pointed to increasing the smartness of data analytics tools and environments in terms of processing and human–system interoperation.***

There were ten needs formulated and grouped into three categories of new functions to expect for the SDATB: (i) interface, (ii) data management, and (iii) smart semantics and procedural reasoning functions. Analyzing the needs related to interfacing revealed they were mainly requesting an intuitive type of interface, easily used and able to help and assist the designer. The intuition is one of the smartness aspects of a system. For data management, the use of context-sensitive and context-adaptive mechanisms within the toolbox is supposed to solve the issues presented by designers and provides more insightful data management approaches. The consideration of context is also an aspect of smartness. Finally, as implied by its name, smart semantics and procedural reasoning function is all about smartness. It can be fulfilled using system learning mechanisms, context information processing, situation awareness, system adaptation capabilities, and so on.

> ***The axiomatic theory fusion methodology overcomes the lack encountered in developing new multidisciplinary theories.***

The theorizing methodologies proposed in the literature could not be directly used in some research areas where a wide range of aspects needed to be considered. In the particular case of knowledge development for a next generation SDATB for white goods designers, theories supporting the development of traditional data analytics tools and software proved to be insufficient. Thus there was a need for developing new theories able to describe the augmented tools as well as their needed functions and computation possibilities. The theorizing approach was developed in this sense. It consisted of deriving comprehensive supporting theories by semantically fusing relevant component theories in a specific topic.

Applying the ATF methodology to our research phenomenon provided an ontological description of what exists and should exist in the development of the SDATB. As presented earlier in this manuscript, ATF was able to make insufficient individual theories more insightful by semantically combining them. The generated propositions in this particular case study form a knowledge platform for the development of the SDATB prototype. It even shaped a skeleton of the toolbox by providing a set of requirements, techniques, and technologies to be considered in the implementation phase. Consequently, ATF was relevant for developing a smart computational system. In conclusion, it can be used in interdisciplinary and multidisciplinary domains.

> ***Theories about (i) designers' explicit needs, (ii) interoperability issues, (iii) decision-making principles, (iv) evolution of data analytics, and (v) enabling technologies are sufficient to build a knowledge platform for a next generation data analytics toolbox for white goods designers.***

While using ATF in developing a knowledge platform for a next generation data analytics toolbox for white goods designers, we semantically combined the five mentioned theories. The outcomes were formulated as a multilevel set of functions and requirements to be implemented for the toolbox. The set of 81 derived propositions was converted into requirements for the SDATB reflecting different components needed for the toolbox development, such as (i) decision-making, (ii) data management, (iii) interfacing, (vi) learning and reasoning, (v) data characteristics, (vi) design issues, and (vii) data analytics tools and techniques. The derived knowledge was brushed up to be used in the conceptualization of the SDATB without requiring more research activities.

6.1.3. Reflections on the conceptualization of a demonstrative smart data analytics toolbox

The third research cycle focused on the conceptualization of the demonstrative data analytics toolbox. Here, the set of SDATB functions was determined based on the outcomes of the ATF methodology application. The obtained functions were filtered to a reduced set used for the conceptualization of a demonstration prototype of the SDATB. While the first activity of the research cycle consisted of building functions originally reported by the activities of the second research cycle, the second activity was to prioritize the functions of the demonstrative concept of the toolbox. Finally, the functional decomposition of the functions of the SDATB demonstration was established.

- **Reflections on the methodology:**

We began the third research cycle by synthesizing the outcomes of the ATF methodology application into possible SDATB functions. The outcomes were checked in the literature to determine the novelty and feasibility of what was reported by the theory. This knowledge aggregation helped refine the findings of previous studies and led us towards the fundaments of the next generation data analytics toolbox. The following step was to determine the overall concept of the SDATB based on the set of needs and expectations of white goods designers we had previously generated. In this stage, the needs and expectations were textually and semantically converted into functions of the toolbox. For the sake of simplification, we focused on the smart basic functions of the toolbox, given their importance in the SDATB conceptualization.

The result was a concept incorporating a reduced set of smart functions as a demonstration. To elaborate the demonstrative conceptual model of the toolbox, we investigated and compared the functions to what exists in the literature to choose the newest and smartest options. After that, the functions were decomposed from high- to low-level decomposition to aid in determining the algorithms to be used in realizing the demonstrative concept. The approach adapted for every function and the functional decomposition were combined to architect the components of the demonstrative prototype. If the time window for the research had been longer, we would have focused on a complete SDATB conceptualization and not a demonstration, and we would have investigated smarter approaches.

- **Reflections on the results:**

The major findings of the activities of the third research cycle are the following:

> ***The outcomes of the second research cycle were enough to build a set of functions for the smart data analytics toolbox.***

In the third research cycle, the objective was the specification of novel smart functions for a next generation SDATB. The QBI executed in research cycle 2 revealed a spectrum of needs and opportunities for the toolbox development. The needs included, but were not restricted to the following: (i) step-by-step assistance, (ii) advice on means of selection, (iii) multifold data visualizations, (iv) multichannel data management, (v) ability to blend datasets, (vi) ability to combine qualitative and quantitative data, (v) permanent accessibility, (vi) adaptation to user, (vii) CBR, and (viii) ability to learn from applications. From another perspective, the theory built with the ATF methodology reflected the need for an SDATB that (i) includes context-driven decision making; (ii) includes proactive decision-making; (iii) includes algorithms able to process complex data; (iv) allows semantic interpretation; (v) blends data and datasets; (vi) merges data streams; (vii) allows high-speed and high-volume storage; (viii) is permanently accessible; (ix) advises designers on their choices; (x) allows CBR; (xi) processes structured, semi-structured and multi-structured data; (xii) proposes solutions to solve difficult design problems; (xiii) predicts future outcomes; and (xiv) derives actionable insights. The formulated needs from both complementary studies give an idea of what is to be expected from the SDATB. A small exercise of sharpening the textual formulation of the needs was enough to determine the functions for the toolbox. Moreover, the functions needed tackled basic, auxiliary, and interface subsystems of the SDATB.

> ***A complete concept of the smart data analytics toolbox includes basic, auxiliary, and interface functions.***

The SDATB should be composed of functions for processing and facilitating the processing of data. These functions are the ones without which the toolbox will not exist. Consequently, they are called basic functions. Within the SDATB, complementary functions are also needed for converting data into knowledge, such as warehousing and operation management. They are called auxiliary functions. Finally, an SDATB must contain interface functions to ensure and facilitate communication with the user.

6.1.4. Reflections on the implementation and functionality validation of the functions of the demonstrative smart data analytics toolbox

The first activity of the fourth research cycle focused on the functional, architectural, algorithmic, and computational considerations in implementing the functions of the demonstrative SDATB. First, we identified, listed, and detailed the algorithms. We then collected, from the literature and the web, information about the prototype-level implementation and the computational techniques to build the SDATB functions. This set of knowledge was enough to realize and implement these functions. In the second activity of this research cycle, we validated the functionality and execution of the algorithms. We built an application case involving enhancement of white goods (a connected washing machine) by product designers and used it in computationally implementing the three functions in Matlab software. This not only allowed us to test the functionality of the modules but also provided information about the feasibility and the computational performance of the components of the demonstrative SDATB.

- **Reflections on the methodology:**

All functions were computationally implemented using the resources offered by Matlab. In terms of the methodology behind this implementation, we reviewed solutions for implementing software and collected information on realizing the algorithms listed in the conceptualization phase. We also explored computational techniques for the prototype-level implementation related to the particular application case of the SDATB.

We focused on software-level implementation of the algorithms for the toolbox functions.

All computational functions were implemented based on the resources offered by Matlab. Although using Matlab facilitated the achievement of all objectives related to the three representative functions, we faced some limitations. Implementation of the basic function was time consuming due to the absence of predefined operations in the software. The deep learning resources offered by the Matlab toolbox proved to be still in a late research phase. It suffers from a lack of smartness in terms of its usage, which meant training the neural network was a time-consuming task. For the auxiliary function, we used Matlab's data analytics tools as the subjects of recommendation. Since it was not automatically generated or retrieved, we had to weight the tools manually. As for the interface function, the image acquisition toolbox suffered from a lack of comprehensiveness; before accessing the camera, we had to specify the image source and the camera resolution. This caused delays in accessing the elements of the toolbox.

The software level implementation allowed us to test the toolbox functions to validate their functionality through operationalization of the case of white goods enhancement by product designers. We were able to check that the modules and their algorithms were working as they should and that the functional requirements were fulfilled. This allowed us to draw conclusions about the feasibility and performance of the algorithms. As mentioned previously, if more time had been dedicated to the research, the implementation could have included the entire SDATB. We would also have considered additional validation aspects, such as structural, applicability, and utility validation. Other types of validation can also be added to this approach, such as properness validations and adopting automation to validation processes.

- **Reflections on the results:**

The following are the major findings of the activities of the fourth research cycle:

> ***The function of merging middle-of-life data streams offers two levels of semantic inferring.***

The two levels of semantic inferring offered by the computational function developed for merging MoLD-Ss are (i) the level of merging, and (ii) the level of decision-making. Technically, the former was fulfilled by (i) employing a proper neural network architecture, (ii) using its attention layer, and (iii) clustering past knowledge with triplet network embedding. The reasoning by the computational function provides opportunities for (i) constructing implicit knowledge graphs, (ii) learning the statistical model, and (iii) separating related and unrelated behavior patterns. The multidimensional latent space captures the similarity considering multiple criteria, and exploration of knowledge clusters can happen in a non-constrained way.

> ***Implementation of a computational recommendation function for choosing task-relevant data analytics tools compensates for product designers' lack of data analytics tools knowledge and reduces the time and effort needed for tool selection.***

The recommendation function helps product designers choose the most appropriate tool for processing data based on their design task. It can interpret a designer's input, propose a description of the problem identified by the designer, reason with the warehoused DATs, and recommend DATs that matching the designer's task at hand. The testing of the functionality proved that the recommended tool is the most appropriate one based on the DTX. This function facilitates the usage of the toolbox by designers. Whether or not the designer has knowledge about or experience with various tools, the toolbox makes the tool choice, saving the designer extra time and effort figuring out which tool may be good for a particular purpose.

Implementation of a smart identification function within the SDATB offers a secure environment that is easy for product designers to access.

The smart identification function features both identification and authentication of the designer. Since two-factor authentication is more secure, we implemented the authentication using biometric identification that comprises both face detection and face recognition. To simplify the process, the computer's integrated or external camera must detect only one face. To augment the security of the SDATB, the smart identification allows recording the history of past identifications of each designer, including the date and time. This can be used to check whether hacking might have occurred. In such a situation, the user has the option to reset the identification, the attendance history, and the password, if used. The biometric identification provides easy access to the toolbox: all the user needs to do is to face the computer's camera.

6.2. Propositions

6.2.1. Scientifically-based propositions

In line with the main objectives of the Ph.D. research work, a set of 10 debatable propositions have been formulated. They capture the main scientific contributions in terms of research methodologies and concrete findings.

- **Propositions of research cycle 1:**

Proposition 1: *The graphical representation offered by VOSviewer stimulated an intensive exploration and a combined qualitative and quantitative characterization of the research phenomenon.*

In the process of initial literature exploration and knowledge aggregation, the VOSviewer software was used to develop a topographic landscape of published works. The research topic and phenomenon have been covered with a wide range of keywords. The graphical representation allowed us to visually observe the density of coverage and the uniformity of the distribution of the publications of the domains of interest. In addition, it provided an overview of the interrelationships among the search terms. Another service offered by the software is grouping keywords in clusters and providing a quantitative characterization of interrelationships between keywords within the same cluster. It also represents the strengths of relationships between the various clusters. Using the VOSviewer allowed us to indicate specific domains of interest for further investigation, as well as to study both the separated and the interconnected keywords. Methodologically, using this tool made it possible to conduct a kind of pathfinding exploration of the related literature.

Proposition 2: *Neither in data analytics nor in design enhancements are middle-of-life data used as extensively and purposefully as they could be.*

MoLD are rich and diverse. In the process of knowledge aggregation reported upon in Chapter 2, we observed that many works dealt with the importance of such data in data analytics processes and how positively MoLD can influence industries and design improvement. On the other hand, the available literature lacked seminal papers reporting on the methods, techniques, and tools to analyze these data. It has been recognized in both academia and industry that the use of MoLD in design practices offers opportunities for understanding the use, performance, failures, and maintenance of products and for improving them based on real-time operation and use data. Products can be adapted or augmented based on how they perform in operation and what the user is doing right, or wrong, with them. Nevertheless, the current situation is that these data are not extensively collected, investigated, or processed to inform design improvements. Moreover, the actual data analytics tools are not designed for analyzing and exploiting MoLD in the context of design enhancements.

- **Propositions of research cycle 2:**

Proposition 3: *The combined inductive study (web-hosted questionnaire-based interrogation) and deductive (axiomatic theory fusion) study not only complemented but also consolidated each other*.

The research approach designed in the third chapter was a multistrand inquiry. It included two major forms of knowledge aggregation and theorization, referred to as inductive study and deductive study, respectively. The first one was intended to investigate the needs of white goods designers through a web-hosted QBI to obtain knowledge from a practical point of view. The second study focused on using the principles of ATF to develop a formal theory of a next generation data analytics toolbox. As reported in Chapter 3 concerning the procedural model of the multistrand inquiry, the first study was a bottom-up knowledge aggregation, whereas the second was a top-down argumentation. The two studies complement each other, since the insights and knowledge generated from the QBI were used as a starting point of the ATF application. After the completion of both studies, we realized that we had detected no contradictions and that the final theory properly framed the theoretical and practical needs of white goods designers.

Proposition 4: *The body of knowledge derived from the complementary inductive and deductive studies provided a sufficient basis to cover the demands for a next generation smart data analytics toolbox.*

As reported in Chapter 3, the knowledge obtained from the inductive and deductive studies circumscribed various needs, expectations, techniques, and technologies for a next generation SDATB. The theoretical and the empirical findings were sorted into clusters. Forming a skeleton body of knowledge for the SDATB, they identified opportunities related to interfacing, decision-making, algorithmic concepts, reasoning and learning, data management, design issues, data characterization, expected outputs, and data analytics methods and techniques. Moreover, the synthesized needs theory and the fused theory obtained using ATF informed not only about the opportunities, functions, techniques, and technologies to be considered in the development of the SDATB but also about what should be avoided and not included in the SDATB. The latter include, for instance, traditional analytics techniques inadequate for analyzing big data generated by smart products and DNNs as being computationally expensive and requiring long training times to recognize patterns in data.

- **Propositions of research cycle 3:**

> ***Proposition 5:*** *The theory synthesized by the axiomatic theory fusion methodology implies generic requirements for the computational functions of the smart data analytics toolbox.*

The theory obtained by the ATF methodology informed the functional expectations for next generation data analytics tools. The concretization and textual specification of these expectations resulted in specific technical requirements for the SDATB. A complex logical analysis was needed to convert the requirements into functions. In the conversion process, the meanings, semantics, and context of each requirement were kept intact from any subjective transformation. In this process of theory fusion, 81 context-sensitive propositions were formulated and converted into the computational functions of the SDATB, which included basic, auxiliary, and interface types of functions. They covered numerous aspects, such as decision-making processes, expected algorithmic concepts, reasoning and learning, data management strategies, interfacing adjustments, data types and characteristics to be supported, design issues to be tackled, analytics techniques and methods to be implemented, and possible outputs to expect from data analytics. The added value offered by the application of the ATF methodology was that a novel set of smart functions were implied for dealing with MoLD and that the smartness of the toolbox was increased as it also happens with smart products.

> ***Proposition 6:*** *Computational functions such as merging MoLD-Ss, recommending choices for task-relevant data analytics tools, and face image-based identification are representative basic, auxiliary, and interface functions for the smart data analytics toolbox.*

An SDATB can be supposed to provide the three smart functions that we chose for implementation. Actually, we considered these functions indispensable in the realization of a smart toolbox. As a basic function, merging MoLD-Ss is needed to provide fast, efficient, and context-dependent processing for this kind of data. The implemented computational function allows processing of MoLD with a minimum number of interactions between the user and the system, and it also provides recommendations about product enhancement opportunities. As an auxiliary function, recommending task-relevant data analytics tools may compensate for designers' lack of knowledge about data analytics tools. This is not considered a basic function because it does not augment the performance of data analytics tools but recommends the best tool to use for a particular design task. As an interface function, the face-image-based identification function offers a trusted identification method and a secure data analytics environment for the designer. This is needed for all software platforms. This is purely an interface function because it allows the designer access to the SDATB in a regulated manner but independent of the data processing and data management functions.

- **Propositions of research cycle 4:**

Proposition 7: *The implementation of the functions of the smart data analytics toolbox was systematized through the integrated process of requirements engineering, functional decomposition, architecting, workflow synthesis, algorithm development, and data construct specification for each computational function.*

We applied a systematic approach to realizing the SDATB functions. First, we harmonized the requirements for the toolbox and converted them into conceptual functions. Actually, a selection of functions were considered for detailed conceptualization and implementation for the demonstrative SDATB. The initial step was the investigation and definition of the objectives, approach, and principles of the chosen functions. Functional decomposition was then completed to break down the main functions into intermediate-level functions and elementary functions, to determine what was needed for their realization. In the next step, the outcomes of the functional decomposition were used for architecting the modules and components for every function and to determine the links between the components. The computational workflow diagrams, including each function, were constructed to show the procedural logic (i.e. the order) and the operational relationships among the algorithms. Based on the outcomes of these processing steps, the standard and proprietary algorithms needed for the implementation of the functions were determined. Finally, the data constructs were specified according to the computational variables included in the algorithms.

Proposition 8: *The algorithms designed and implemented separately to provide recommendations for choosing task-relevant data analytics tools are not smart in themselves, but their combination made them function smartly.*

In implementing the function to provide recommendations to support choosing task-relevant data analytics tools, we built eight algorithms: (i) retrieve DST with minimum distance to DT given by designer, (ii) build DSTs' vectors, (iii) build vector DAT, (iv) calculate distance between DSTs and DAT vectors, (v) sort DATs, (vi) retrieve DAT vectors most similar to DST vector, (vii) rank DATs, and (viii) retrieve best finite set of DATs. All of these algorithms can be considered traditional because they do not include reasoning or learning capabilities, and because their inputs are all pre-programmed. Nevertheless, together they were able to deliver smart functioning from the perspective of the user, as based only on the user's design task, the implemented function can propose the best DAT for this task. The system is traditional, but its manifestation to the user is smart.

Proposition 9: *The functional validation also provides information about the computability and the performances.*

The results of the functional validation of the implemented functions reported in Chapter 5 were used to evaluate the functionality of the algorithms in the reference application case of enhancement of a washing machine by product designers based on MoLD. Since the pseudo-coding and the implemented algorithms did not show errors, we can conclude that the modules were computationally correct. Moreover, the fast and reliable computation reflected the high performance of the algorithms as well as the choice of the implementation. In this sense, the feasibility, several performance aspects related to the computation, and the functionality of the modules of the demonstrative SDATB were tested and validated.

Proposition 10: *The recommendation to support choosing task-relevant data analytics tools is quasi-application-independent.*

The principal of the recommendation function that supports choosing task-relevant data analytics tools is the following: recommending an item to a user whose preferences are unknown. Based on user input requested by the system, a filtering is applied to offer a ranked, limited set of items that can meet and satisfy the user's expectations. The principle of the recommendation function is independent of the application and the user. We applied it in the particular case of white goods enhancement by product designers. Otherwise, the system can be built independently of the application. The contextualization of the recommendation function depends then on changing the input and the items.

Proposition 11: *Combining multimodal data merging and searching for similar anomalies allows propositional reasoning in a smart data analytics toolbox.*

Multimodal data merging and a search for similar anomalies solve separate tasks. To develop the merging MoLD-Ss recommendation function, they were combined, since sensors provide prerequisites for recommended design enhancement opportunities based on reasoning with past cases. The incorporated knowledge in the database contains possible actions for the designer to undertake. A developed similarity search model is capable of recommending the most appropriate one to take into account. Moreover, to adopt the action, the designer needs to understand the reason and motivation behind it, and information needs to be provided for the designer. The data fusion step estimates the importance of sensors and proposes which ones are responsible for detected anomalies. This reasoning problem is solved by suggesting the most relevant action and justifying it with sensor-related anomalies.

6.2.2. Socially-based propositions

Proposition 11: *Continuous analysis of MoLD using appropriate smart data processing tools will create a revolution in all industries.*

Proposition 12: *Semantic interpretation and analysis of data is a major step towards designers' cognitive satisfaction.*

Proposition 13: *A reliable recommendation function to support the selection of task-relevant data analytics tools makes the experience of using* *complex multifunctional data analytics toolboxes more pleasing for product designers.*

Proposition 14: *The balanced needs and informed expectations of the users must be considered in generating up-to-date data analytics tools.*

Proposition 15: *Combining both qualitative and quantitative approaches for the same purpose provides robust knowledge that cannot be obtained when they are applied separately*.

6.2.3. Self-reflective propositions

Proposition 16: *Enthusiasm about research makes you start a Ph.D., while realism makes you finish it.*

Proposition 17: *Having a "big mouth" is the best compliment a researcher can receive.*

6.3. Recommendations for future work

6.3.1. Short-term follow-up research

Four years have been spent on this research topic, related to which many new and interesting scientific findings were obtained. Despite this, this work can be improved, and several short-term follow-up research activities can be seen. Based on the studies summarized in this dissertation, these may include the following:

- An all-embracing implementation of the SDATB could not be the target of this work. However, a full-fledged implementation is deemed not only possible but also necessary for professional and commercial reasons.
- The implemented functions can be improved, considering the recognized limitations. More sophisticated smart algorithms can be considered to make these functions more intelligent and autonomous.
- Concerning a full-scale implementation, efforts can be dedicated to the synergistic integration of all conceptualized and additional modules in the SDATB.
- Concerning the validation of the usability and utility, more aspects of validation can be considered in studying the use and impact of the implemented functions.

The starting point for the follow-up investigations and the full-featured implementation can be the (incomplete) set of functions presented in the conceptualization chapter. These functions include: (i) adaptation of the toolbox to the user, (ii) learning from toolbox applications, (iii) affording permanent accessibility of the toolbox, and (iv) offering step-by-step assistance all throughout the SDATB usage. In this sense, future research activities should extensively address the specific needs of white goods designers presented in our background study. Extending the SDATB to a context-sensitive product enhancement recommendation system would also be desirable. Moreover, the smart reasoning and learning mechanisms of the SDATB may also be enhanced. These operational features are needed to be able to address meanings and semantic interpretations in the process of analyzing design tasks. This will make it easier for designers not only to analyze data but also to interpret them, to infer from them, and to make decisions based on them. The smartness of the SDATB will help in keeping up with the fast-developing smartness of consumer durables and other products, which are progressing towards smart cyber-physical systems.

6.3.2. Long-term research opportunities

One long-term research opportunity is the augmentation of the principles of the SDATB by allowing (i) historic operation use situation/change analysis, (ii) multisource output data fusion, (iii) simulation-based prediction and forecasting, (iv) visually based feature learning, (v) event occurrence monitoring ("watchdogging"), (vi) semantic association graph building, and (vii) context-based advising. The functions of the SDATB will purposefully include semantic data modeling and reasoning tools. This toolbox will be developed to keep up with next generations of products, which will be equipped with the capability of generating real-time MoLD, communicating about their objectives and states, building awareness and reasoning about the states and objectives, and adapting themselves toward optimal performance. On the other hand, the considered elements of smartness are supposed to allow designers making the products more autonomy and the ability to carry out part of the data analytics quasi-automatically. This provides an opportunity for using multiple products as interacting agents contributing to multi-aspect enhancements. This also means that designers will use the SDATB in cases of large-scale and multisource-dependent data processing.

Another opportunity will be to consider sharing smart data analytics functions between the concerned products themselves and the product data analytics environment. The reason for doing this is that products are becoming equipped with more and more smart capabilities, which enable them to gather and process data by themselves in run time and self-adapt themselves according to the operational conditions and altering objectives. What it means is that products can take over a part of the function of the SDATB. This is a new research phenomenon and challenge that needs extensive study. Putting everything together, research into and development of an SDATB able to assist in anticipating real-life use patterns and in decision-making about product enhancement is relevant not only for the scientific community but also for several segments of the making industry.

Summary

Exemplifying smart functions for a next generation data analytics toolbox

Background of the research

There is rapid development of software technologies focusing on exploration of big data. Consequently, companies need to better manage the whole lifecycle of their products. However, most of the efforts have been dedicated to supporting beginning-of-life and end-of-life models and activities. Fewer efforts have been made to exploit middle-of-life data (MoLD) and to create value and knowledge based on this kind of data. Thanks to new information technologies such as sensors and smart tags, the information of the middle-of-life phase can now be identified, tracked, and collected. Still, there is a lack of tools to support design and servicing decision-making that uses this particular type of data.

Effective processing of MoLD is not only an academic challenge but also a useful asset for industry, since logistics, operation, and maintenance activities are located in the middle-of-life stage. It is important for product developers and companies to find a way to analyze how their products are used by different customers in different environments and circumstances. This will give them rich insights into how to transform use patterns to product enhancements. MoLD are rich data. They can be aggregated through field observations and surveys of users, or by studying failure log files and maintenance reports, or from relevant web resources such as social media and user forums. Alternatively, they can be elicited directly from products by sensors or self-registrations. However, due to the dynamic change of sensor data, the large volumes of data aggregated over time, and the unknown nature of data patterns, it is unfortunately not straightforward to perform effective data analysis using traditional data processing tools. The main challenge is to find ways to effectively use data analytics techniques in purposeful combinations, depending on the application contexts and the specific objectives of product designers.

Research phenomenon

Existing data analytics tools present a number of challenges: (i) exploiting and controlling rapidly changing patterns of use, (ii) dealing with generic tools in specific product development cases, (iii) covering all data transformation steps, (iv) combining and integrating outcomes, and (v) interpreting the meanings of the outcomes in the context of the product development tasks at hand. Although numerous data analytics software tools have been developed for extracting and exploiting data, analytics methods and tools in product enhancement are still in a premature stage. Yet, no concrete solutions are included in commercialized data analytics systems. Moreover, there is a lack of computational mechanisms to support decision-making and servicing, as well as a lack of theories explaining how to select, combine, and deploy existing mechanisms and software tools in cases of product-use data and MoLD.

The overall objective of this thesis was to cover the lack of data analytics tools designers can use to process MoLD. Towards this end, the research goal was to first generate requirements and fundamentals for a new smart data analytics toolbox (SDATB) able to overcome the issues related to traditional commercial tools, then convert them into concrete functions to be implemented in the future. Therefore, the guiding research question has been formulated as follows: "What functions are to be included in a next generation smart data analytics toolbox to support product designers in enhancing products and services based on MoLD?"

Methodology and content of research cycle 1

The overall objective of the first research cycle was knowledge aggregation concerning data analytics tools in the context of supporting product and service enhancement based on MoLD. This research cycle was methodologically framed as a research in design context, divided into an explorative part and a confirmative part. To aggregate the state-of-the-art knowledge related to our topic, the following domains of interest were surveyed: (i) nature of data, (ii) data transformation steps and techniques, (iii) data transformation means (tools and packages), and (iv) applications of data analytics. These domains were investigated to determine their current state, the knowledge gaps, and limitations of existing data analytics tools and techniques in the context of product enhancement by product designers using MoLD. The findings of the literature study were synthesized, including the limitations of existing data analytics tools and packages. These limitations led to the identification of opportunities for data analytics tools development.

Methodology and content of research cycle 2

The second research cycle focused on the elaboration of a set of fundamentals and requirements needed for the realization of a next generation data analytics toolbox. To this end, we undertook two main activities: (i) a web-based interrogation was conducted to understand designers needs related to data analytics practices and tools. Then, (ii) it was complemented by the synthesis of theories needed for data analytics tools development. This synthesis involved merging theories using the methodology of axiomatic theory fusion (ATF) developed for this concrete purpose. The methodological framing applied in both activities was a research in design context. According to its principles, two successive phases took place, namely explorative and confirmative phases. In the first activity, the exploration consisted of a web-hosted questionnaire-based interrogation and a literature study, to investigate designers needs and to derive knowledge from the literature. The outcomes of the two studies were compared in the confirmative phase. For the second activity, the explorative phase consisted of investigating existing theories related to the development of data analytics tools and building the methodology to combine them in this specific context. The confirmative phase applied the methodology developed in the context of the study to create a new theory consolidating requirements, principles, and functions of a new SDATB taking into consideration practical designers' needs. At the end of this research cycle, enough knowledge had been obtained to draw conclusions about fundamentals and requirements for the SDATB.

Methodology and content of research cycle 3

The third research cycle focused on the ideation and the conceptualization of a demonstrative SDATB. Consequently, this research cycle was framed according to the design inclusive research methodology. Research activities of this research cycle were organized in three phases (i) explorative, (ii) constructive, and (iii) confirmative. The explorative phase concentrated on the exploration of knowledge and enabling technologies for the toolbox conceptualization. The outcomes of the previous research cycle were synthesized to produce the requirements and fundamental concepts related to the SDATB. The constructive phase established a comprehensive conceptual model of a demonstrative SDATB. Accordingly, the functions of the toolbox were investigated, the concept was generated, the functional decomposition established, and the high-level architecture built. These functions are as follows: (i) recommendation for merging of middle-of-life data streams (MoLD-Ss), (ii) recommendation of task-relevant data analytics tools, and (iii) smart user identification. The confirmative phase focused on testing the feasibility of all computational constructs of the SDATB.

Methodology and content of research cycle 4

The fourth research cycle was dedicated to implementing and validating the functionality of the functions of the demonstrative SDATB. This research cycle was framed according to a design inclusive research methodology, with an operative design research flavor. It included three phases: (i) explorative, (ii) constructive, and (iii) confirmative phases. The explorative phase focused on (i) determining and detailing the algorithms to be used for the implementation of the demonstrative SDATB, (ii) reviewing existing solutions that enable the implementation of data structures and algorithms for each functionality, (iii) collecting information about the fundamentals that support designing the needed algorithms, and (iv) investigating the logical and computational techniques for the prototype level implementation with the application case of white goods designers using MoLD to enhance a washing machine. The constructive phase focused on the software-level implementation of modules and algorithms of the demonstrative SDATB. Finally, the confirmative phase validated the feasibility of the implemented modules of the component functions in Matlab software.

Main findings of the research

The main findings of this research project can be summarized as follows: In the first research cycle, the use of the software VOSviewer helped portray a topographic landscape of the state of the art related to our research topic. It exposed the domains of interest, clustered keywords, and provided a quantitative characterization of interrelationships between keywords within the same cluster. It represented clearly the strengths of relationships between clusters. Using VOSviewer directed us towards a clear path for the literature investigation. In the deep investigation of the literature, we observed that MoLD is important in data analytics processes, although there are a limited number of papers reporting on methods, techniques, and tools to analyze MoLD. The use of MoLD in design practices offers opportunities to understand the use, performance, failures, and maintenance of products and to improve them based on real-time generation of user data. Products can be augmented based on what the user is doing right, or wrong, with them, but the current situation is that these data are not extensively investigated or used in processing design improvements. Additionally, the existing data analytics tools are not designed for analyzing MoLD in design enhancement contexts.

In the second research cycle, the investigation of the needs of white goods designers through a web-hosted questionnaire-based interrogation was useful in obtaining knowledge from a practical point of view, while the development and use of ATF principles to generate a formal theory of a next generation data analytics toolbox completed the picture of what is needed for a next generation SDATB. The two studies complement each other, since the insights and knowledge generated from the web-hosted questionnaire-based interrogation were used as the starting point of the ATF application. After the completion of both studies, we realized we had detected no contradictions and that the final theory reported on theoretical and practical needs of white goods designers. At the end of this cycle, we had a set of needs, expectations, techniques, and technologies for a next generation SDATB. In addition, these findings were contained into clusters, forming a skeleton of the SDATB. Finally, the outcomes of complemented studies reflect on what should *not* be included in the SDATB.

In the third research cycle, the theory synthesized by application of the ATF methodology clearly identified what should be expected from a next generation SDATB. The textual formulation of this theory included the requirements for the toolbox without any subjective conversion. The reformulation was needed only to convert the requirements into functions to make the message of each functionality clear. From the set of possible smart computational functions, we chose three functions for further detailing and implementation: (i) merging of MoLD-Ss, (ii) recommendation of task-relevant data analytics tools, and (iii) smart user identification. The first function was chosen because the primary objective of the SDATB is to analyze MoLD. We decided to combine it with two needs: (i) the need for semantic interpretation of data analytics outputs and (ii) the need for merging different data streams from multiple sources. With this function, we semantically merge MoLD-Ss, and based on the merging results, we offer recommendations to the designer as an action plan of what needs to be done with the product. The second function was directly extracted from the most needed functionality as identified by designers, which was getting advice on what tool to use for various analyses. This function will offer to the designer a recommendation about the best tool to use for the design task at hand. The last function was chosen because it is an important requirement that the SDATB offer a safe and secure analysis environment for the user to protect the user's data and knowledge. These functions were conceptualized, decomposed, and architected in this research cycle.

In the fourth research cycle, the functional decomposition of the component functions, their architecture, the algorithm specifications and designs, and the workflow diagrams were important steps elaborated and synthesized to build the algorithms needed for implementing the functions. For the merging MoLD-Ss recommendation, ten algorithms were designed, and others were used from the literature. The algorithms included a certain level of smartness, since they allowed reasoning and learning with and from actual data streams as well as from past anomalies, and the function proposes a recommendation based on the context of the analyzed data streams. We built eight algorithms for the function that provided recommendations to support the choice of task-relevant data analytics tools. The algorithms for this function do not include any reasoning or learning capabilities; nevertheless, they were able to deliver smart functionality from the perspective of the user, as based only on the design task, the implemented functionality can propose the best data analytics tool for that task. The system is traditional, but its manifestation to the user is smart. For the smart user identification function, some of the algorithms were built and others were adapted from existing algorithms. The implemented functions improve the smartness of the system and provide advantages to the user, such as (i) semantically merging data streams and offering a context-based recommendation for the designer on what action to take to enhance the product (for the first function), (ii) compensating for designers' lack of knowledge about data analytics tools (for the second function), and (iii) offering a secure environment (for the third function). Finally, validating the functionality of the implemented functions using the reference application case of product designers using MoLD to enhance a washing machine detected no errors in the algorithms, and the modules were computationally correct. The validation process also made clear the reasonable speed and reliability of the computation.

Samenvatting

Het voorbeeld van slimme functies voor een volgende generatie data-analyse toolbox

Achtergrond van het onderzoek

Softwaretechnologieën die zich richten op het verkennen van big data zijn volop in ontwikkeling. Daarom dienen bedrijven de volledige levenscyclus van hun producten beter te beheren. De meeste inspanningen zijn echter gericht op het ondersteunen van modellen en activiteiten over de begin- en eindfase van producten. Er zijn minder inspanningen geleverd om middle-of-life data te benutten en om, op basis van dit soort gegevens, waarde/kennis te creëren. Dankzij nieuwe informatietechnologieën zoals sensoren en smart tags kan de informatie van de middelste levenscyclus nu worden geïdentificeerd, opgevolgd en verzameld. Toch is er een gebrek aan instrumenten om de besluitvorming over het ontwerp en het onderhoud van producten met behulp van dit specifieke type gegevens te ondersteunen.

Effectieve verwerking van middle-of-life data (MoLD) is niet alleen een academische uitdaging, maar vormt ook een waardevolle troef voor de industrie, aangezien de logistieke, operationele en onderhoudsactiviteiten zich in het midden van de levenscyclus bevinden. Voor productontwikkelaars en bedrijven is het van belang te begrijpen hoe hun producten door verschillende klanten worden gebruikt in verschillende omgevingen en omstandigheden. Dit zal hen waardevolle inzichten bieden over hoe zij gebruikspatronen kunnen omzetten naar productverbeteringen.

MoLD zijn waardevolle gegevens. Ze kunnen worden verzameld door middel van waarnemingen in de praktijk en bevragingen van gebruikers, door het bestuderen van storingslogbestanden en onderhoudsrapporten, of via relevante internetbronnen zoals sociale media en gebruikersfora. Ook kunnen MoLD direct uit producten worden gehaald door sensoren of zelfregistraties. Door de dynamische verandering van sensorgegevens, de grote hoeveelheden gegevens die in de loop van de tijd worden verzameld en de onbekende aard van gegevenspatronen, is het helaas niet eenvoudig om een effectieve gegevensanalyse uit te voeren met behulp van de bestaande traditionele dataverwerkingstools. De belangrijkste uitdaging is om manieren te vinden om effectief gebruik te maken van data-analysetechnieken in doelgerichte combinaties, afhankelijk van de toepassingscontext en de specifieke doelstellingen van productontwerpers.

Onderzoek naar het fenomeen

De bestaande instrumenten voor data-analyse stellen ons voor een aantal uitdagingen: (i) het benutten en beheersen van snel veranderende gebruikspatronen, (ii) het omgaan met generieke instrumenten in specifieke gevallen van productontwikkeling, (iii) het behandelen van alle stappen van in het omzetten van gegevens, (iv) het combineren en integreren van de resultaten, en (v) het interpreteren van de betekenis van de resultaten in de context van de desbetreffende productontwikkelingstaken. Hoewel er tal van software-instrumenten voor data-analyse zijn ontwikkeld voor het verzamelen en benutten van gegevens, bevinden analytische methoden en instrumenten voor productverbetering zich nog in een vroegtijdig stadium. Momenteel zijn er geen concrete oplossingen ingebouwd in commerciële data-analysesystemen. Bovendien is er een gebrek aan rekenkundige mechanismen om de besluitvorming en het onderhoud te ondersteunen. Daarnaast is er ook een gebrek aan theorieën die verklaren hoe wij bestaande mechanismen en software-instrumenten kunnen selecteren, combineren en inzetten in het geval van productgebruiksgegevens / MoLD.

Deze thesis had voornamelijk tot doel om het gebrek aan data-analysetools voor het verwerken van MoLD door ontwerpers te behandelen. Het onderzoeksdoel was hierbij om in eerste instantie vereisten en basisprincipes te identificeren voor een nieuwe slimme data-analyse toolbox die in staat is om de problemen van traditionele commerciële tools te overwinnen, en die vervolgens om te zetten in concrete functies die in de toekomst moeten worden geïmplementeerd. Daarom is de leidende onderzoeksvraag als volgt geformuleerd: "Welke functies moeten worden opgenomen in een volgende generatie slimme data-analyse toolbox om productontwerpers te ondersteunen bij het verbeteren van producten en diensten op basis van MoLD?".

Methodologie en inhoud van onderzoekscyclus 1

De eerste onderzoekscyclus had hoofdzakelijk tot doel kennis te verzamelen over data-analysetools in het kader van de ondersteuning van product- en serviceverbetering op basis van middle-of-life-data. Deze onderzoekscyclus werd methodologisch uitgevoerd als een research in design context (RDC), opgesplitst in een verkennend en een bevestigend deel. Om de meest actuele kennis over ons onderwerp te verzamelen, hebben wij de volgende domeinen onderzocht: (i) de aard van de gegevens, (ii) de stappen en technieken in het omzetten van gegevens, (iii) de instrumenten voor het omzetten van gegevens (tools en pakketten), en (iv) de data-analyse toepassingen. Wij hebben deze domeinen onderzocht om de huidige stand van zaken, de kenniskloof en de beperkingen van bestaande data-analysetools en technieken te bepalen in de context van productverbetering door productontwerpers op basis van MoLD. Wij hebben een synthese van de bevindingen uit de literatuurstudie gemaakt, met inbegrip van de beperkingen van bestaande data-analysetools en -pakketten. Deze beperkingen hebben geleid tot het identificeren van kansen voor de ontwikkeling van data-analysetools.

Methodologie en inhoud van onderzoekscyclus 2

De tweede onderzoekscyclus was gericht op de uitwerking van een set aan basisprincipes en vereisten die nodig zijn voor de ontwikkeling van een nieuwe generatie data-analyse toolbox. Daarom hebben wij de volgende twee hoofdstappen uitgevoerd: (i) er werd een online bevraging uitgevoerd om inzicht te verwerven in de noden van ontwerpers met betrekking tot de praktijken en tools van data-analyse. Vervolgens (ii) werd dit aangevuld met een synthese van theorieën die nodig zijn voor de ontwikkeling van data analyse tools. Deze synthese bestond uit het samenvoegen van theorieën aan de hand van de axiomatische theoriemethode ontwikkeld voor dit concrete doel. Voor beide stappen werd het methodologisch kader van RDC toegepast. Volgens de principes van RDC vonden er twee opeenvolgende fasen plaats, namelijk een verkennend en een bevestigend deel.

Voor de eerste stap bestond de verkenning uit een web-hosted questionnaire-based interrogation en een literatuurstudie om de noden van ontwerpers te onderzoeken en kennis uit de literatuur af te leiden. De resultaten van beide studies werden in de bevestigende fase met elkaar vergeleken. Voor de tweede stap bestond de verkennende fase uit het onderzoeken van bestaande theorieën met betrekking tot de ontwikkeling van data-analysetools en het bouwen van de methodologie om die theorieën in deze specifieke context te combineren. De bevestigende fase bestond erin de methodologie ontwikkeld in het kader van de studie toe te passen om een nieuwe theorie te creëren die de vereisten, basisprincipes en functies van een nieuwe slimme data-analyse toolbox (SDATB) integreert, rekening houdend met de praktische noden van productontwerpers. Aan het einde van deze onderzoekscyclus hebben wij voldoende kennis verworven om tot een conclusie te komen over de basisprincipes en vereisten voor de SDATB.

Methodologie en inhoud van onderzoekscyclus 3

De derde onderzoekscyclus was gericht op de ideeënvorming omtrent en de conceptualisering van een voorbeeld van SDATB. Deze onderzoekscyclus werd dan ook opgebouwd volgens de design inclusive research (DIR) onderzoeksmethodologie. De onderzoeksactiviteiten van deze onderzoekscyclus werden volgens drie fasen georganiseerd, namelijk (i) een verkennende, (ii) constructieve en (iii) een bevestigende fase. De verkennende fase was gericht op het identificeren van de kennis en de faciliterende technologieën voor de conceptualisering van de toolbox. Hierbij werd een synthese gemaakt van de resultaten van de vorige onderzoekscyclus om te komen tot de vereisten en fundamentele concepten voor een voorbeeld van SDATB. De constructieve fase was gericht op de uitwerking van een uitgebreid conceptueel model voor een voorbeeld van SDATB. In dat opzicht werden de functies van de toolbox onderzocht, werd het concept uitgewerkt, werd de functionele ontleding vastgelegd en werd de architectuur op hoog niveau gebouwd. De betreffende functies zijn: (i) het samenvoegen van aanbevelingen uit middle-of-life datastromen, (ii) de aanbevelingen van taakrelevante data-analysetools en (iii) slimme gebruikersidentificatie. De bevestigende fase was gericht op het testen van de haalbaarheid van alle rekenkundige constructies van de SDATB.

Methodologie en inhoud van onderzoekscyclus 4

De vierde onderzoekscyclus was gewijd aan de implementatie en de validatie van de functionaliteit van de functies van een voorbeeld van SDATB. Deze onderzoekscyclus werd uitgevoerd volgens de DIR methodologie met een operative design research element. Het omvatte drie fasen, namelijk (i) een verkennende, (ii) constructieve en (iii) bevestigende fase. De verkennende fase was gericht op (i) het bepalen en uitwerken van de algoritmen die gebruikt moeten worden voor de implementatie van een voorbeeld van SDATB, (ii) het beoordelen van bestaande oplossingen die de implementatie van datastructuren en algoritmen voor elke functionaliteit mogelijk maken, (iii) het verzamelen van informatie over de basisprincipes die de ontwikkeling van de benodigde algoritmen ondersteunen, en (iv) het onderzoeken van de logische en rekenkundige technieken met betrekking tot de prototype-implementatie. Hiervoor gebruiken wij de use case van een wasmachine optimalisatie door een ontwerper van huishoudapparaten met behulp van MoLD. De constructieve fase was gericht op de softwarematige implementatie van modules en algoritmen van een voorbeeld van SDATB. Tot slot was de bevestigende fase gericht op de validatie van de haalbaarheid van de geïmplementeerde modules van de componentfuncties in de Matlab-software.

Belangrijkste bevindingen van het onderzoek

De belangrijkste bevindingen van dit onderzoeksproject kunnen als volgt worden samengevat: In de eerste onderzoekscyclus hielp het gebruik van de VOSviewer software bij het schetsen van een totaalbeeld over de actuele kennis betreffende ons onderzoeksonderwerp. De eerste onderzoekscyclus identificeerde de onderzoeksdomeinen, clusterde de sleutelwoorden en zorgde voor een kwantitatieve typering van de onderlinge relaties tussen sleutelwoorden van dezelfde cluster. Op die manier werd de sterke relatie tussen verschillende clusters duidelijk. Het gebruik van VOSviewer heeft een duidelijke richting voor het literatuuronderzoek aangegeven. In het diepgaand literatuuronderzoek werd vastgesteld dat MoLD belangrijk zijn in data-analyseprocessen. En dit, hoewel er slechts een beperkt aantal papers gewag maken van de methoden, technieken en tools om deze te analyseren. Het gebruik van MoLD in de ontwerppraktijk biedt kansen om het gebruik, de prestaties, de storingen en het onderhoud van producten beter te begrijpen en te verbeteren op basis van real-time productie van gebruikersgegevens. Producten kunnen worden verbeterd op basis van wat de gebruiker er goed of fout mee doet, maar in de huidige situatie worden deze gegevens niet uitgebreid onderzocht of ingezet voor ontwerpverbeteringen. Bovendien zijn de huidige data-analysetools niet ontworpen voor het analyseren van MoLD in de context van ontwerpverbeteringen.

In de tweede onderzoekscyclus was het onderzoek naar de noden van ontwerpers van huishoudapparaten - door middel van een online vragenlijst - nuttig om kennis te verwerven vanuit een praktisch oogpunt. Daarnaast heeft de ontwikkeling en het gebruik van axiomatische theorieprincipes met het oog op de uitwerking van een formele theorie een volledig beeld geschetst van wat nodig is voor de ontwikkeling van een nieuwe generatie data-analyse toolbox. Beide studies vullen elkaar aan, aangezien de inzichten en de kennis die voortkwamen uit de QBI gebruikt werden als uitgangspunt voor de toepassing van de axiomatische theorie. Na het voltooien van beide studies werd duidelijk dat er geen tegenstrijdigheden zijn ontdekt en dat de uiteindelijke theorie een verklaring geeft voor de theoretische en praktische noden van ontwerpers van huishoudapparaten. Dit heeft uiteindelijk geleid tot een set van noden, verwachtingen, technieken en technologieën voor een nieuwe generatie SDATB. Daarnaast werden deze bevindingen opgenomen in clusters die de architectuur vormen voor de SDATB. Ook de resultaten van aanvullende studies reflecteren over wat niet in de SDATB moet worden opgenomen.

In de derde onderzoekscyclus gaf de synthese van de theorie, na de toepassing van de axiomatische theoriemethode, duidelijk aan wat er kan verwacht worden van een nieuwe generatie SDATB. De tekstuele formulering ervan bevatte de vereisten voor de toolbox zonder enige subjectieve conversie. De herformulering was enkel nodig om de vereisten in functies om te zetten die de boodschap van elke functionaliteit duidelijk maken. Uit de set van mogelijke slimme rekenfuncties kiezen we drie functies voor verdere uitwerking en implementatie: (i) de samenvoeging van middle-of-life datastromen, (ii) de aanbevelingen van taakrelevante data-analysetools, en (iii) slimme gebruikersidentificatie. De eerste functie werd gekozen omdat het primaire doel van de SDATB het analyseren van middle-of-life data is. We hebben besloten om het te combineren met twee noden, namelijk (i) de nood aan semantische interpretatie van de data-analyseoutput, en (ii) de nood aan het samenvoegen van verschillende datastromen afkomstig uit meerdere bronnen. Door deze functie voegen we semantisch middle-of-life datastromen samen en op basis van die resultaten geven we vervolgens aanbevelingen aan de ontwerper in de vorm van een plan van aanpak voor het product. De tweede functie werd rechtstreeks afgeleid uit de door ontwerpers meest gevraagde functionaliteit, met name advies over welke tool te gebruiken voor verschillende analyses. Die functie zal de ontwerper adviseren over welke tool best te gebruiken voor de gegeven ontwerpopdracht. De laatste functie werd gekozen omdat het een belangrijke vereiste is voor de SDATB om een veilige analyseomgeving te bieden voor de gebruiker om zijn gegevens en kennis te beschermen. Deze verschillende functies werden in deze onderzoekscyclus geconceptualiseerd, ontleed en opgebouwd.

In de vierde onderzoekscyclus werden de functionele ontleding van de componentfuncties, hun architectuur, de specificatie en het ontwerp van de algoritmen, evenals het workflow schema als belangrijke stappen uitgewerkt en samengevat om vervolgens de algoritmen te ontwikkelen die nodig zijn voor de implementatie van de functies. Voor het samenvoegen van aanbevelingen uit middle-of-life datastromen werden er tien algoritmes ontworpen en werden de overige uit de literatuur gebruikt. De algoritmen bevatten een zekere mate van intelligentie omdat ze in staat zijn om te redeneren en te leren uit en van actuele datastromen en anomalieën uit het verleden (CBR) en vervolgens een aanbeveling te doen in de context van de geanalyseerde datastromen. Voor de aanbevelingen die de keuze van taakrelevante data-analysetools ondersteunen, werden acht algoritmen gebouwd. Deze algoritmen bevatten geen redeneer- of leermogelijkheden, maar zijn wel in staat om een slimme functionaliteit te leveren vanuit het gebruikersperspectief. Gezien het enkel gebaseerd is op de ontwerpopdracht, is de geïmplementeerde functionaliteit in staat om de best passende DAT voor te stellen tot tevredenheid van zijn design taak. Het systeem is misschien wat traditioneel, maar de manifestatie ten aanzien van de gebruiker is wel slim. Voor de slimme gebruikersidentificatie werden sommige algoritmen gebouwd en andere aangepast op basis van bestaande algoritmen. De geïmplementeerde functies verbeteren de intelligentie van het systeem en bieden voordelen voor de gebruiker, zoals (i) het semantisch samenvoegen van datastromen en het bieden van een gecontextualiseerde aanbeveling aan de ontwerper over de te ondernemen actie om het product te verbeteren (voor de eerste functie), (ii) het opvangen van het gebrek aan kennis die ontwerpers hebben met betrekking tot data-analysetools (voor de tweede functie), en (iii) het bieden van een veilige omgeving (voor de derde functie).

Tot slot bleek uit de validatie van de functionaliteit van de geïmplementeerde functies in de use case van een wasmachine optimalisatie door productontwerpers op basis van middle-of-life data dat er geen fouten werden gedetecteerd in de algoritmen en dat de modules rekenkundig correct waren. Het validatieproces verschafte ons ook informatie over de redelijke snelheid en betrouwbaarheid van de berekening.

Appendix 1

Web-hosted questionnaire-based interrogation

Title: Relationship among data analytics tools and white goods designers

Subjects: White goods designers and non-data analysts.

Targeted application field: Development and enhancement of white goods.

Type of research: Remotely conducted structured deep interrogation based on largely predefined questionnaire.

Stata information: Thanks to the wide range and variety of white goods products and also thanks to their evolutions towards smart and autonomous products, in our interrogation we are targeting design groups and individuals from different companies and different domains of expertise to have insights on their design process.

We mean by white goods mainly household appliances such as: refrigeration equipment, different types of cooker and microwave ovens, washing equipment, drying equipment, air conditioners, etc.

Because the innovations and designs change whiting the same company according to consumers' cultures, so the first criterion of sampling will be the continent the designer designs products for. Then other criteria will occur such as: dividing the designers according to their generation, years of expertise, job position and the specific category of products they are responsible of designing.

Sampling and piloting: All information about them is given in chapter 2

Knowledge to be extracted: We want to detect when in the design process designers use product data generated from their products as well as from different other sources allowing data collection and how it is processed using data analytics tools. We are aiming at extracting designers' needs and preferences regarding the outputs of data analytics tools to be able to propose a solution fitting their expectations and containing features for easy and guarantied product and service enhancement in line with new generations of white goods products.

Objective: Determine what data analytics approaches and tools subjects the designers currently use? for what purposes in the target application field? what they found useful? what else they need? and what their expectations are in terms of data analytics tools?

Exploitation: To define functions of a data analytics toolbox to extract valuable knowledge from big data to be used by designers in the process of development and enhancement of their products and services.

Introduction:

The research we are conducting focuses on using data extracted in the middle-of-life (servicing, operation, use, failure, etc.) of white good products for their design enhancement. Our main research question is: What designers need, prefer, and/or miss regarding the outputs of data analytics tools they currently use or would use in the future in the context of possible product improvements? We are seeking for a practical solution that supports the generation of design knowledge for realization of new products and value adding services by monitoring the real-use of products and services in operation, as well as the user feedback on social media. We also pursue extending product and service lifespan and to optimize the use of the necessary resources all along their lifecycle. In this study, we concentrate on computer supported analytics of big data generated in the middle-of-life processes of white goods. In order to understand designers needs and challenges as well as their design intents and expectations with regards to affordances and services provided by software tools and packages our questionnaire is divided to three main parts: (i) subject information, (ii) product/tasks information, (iii) knowing/using big data, (iv) knowing/using data analytics tools, (v) needed services/functions, (vi) additional information/suggestions.

Time duration: Between 10 to 20 min.

The questions as sent to white goods designers:

Subject information

Q1: Are you engaged with designing white goods?

- □ yes
- □ no
- □ indirectly

Q2: What is your highest degree of education?

- □ secondary school
- □ high school

- ☐ university bachelor
- ☐ university master
- ☐ university Ph.D.
- ☐ other, please specify…………………………………………………………...

Q3: How many years of product design experience do you have?
- ☐ 0 – 1 year
- ☐ 1 – 3 years
- ☐ 3 – 6 years
- ☐ 6 – 10 years
- ☐ 10 – 20 years
- ☐ 20 – 40 years
- ☐ more than 40 years

Q4: How long do you work for your current company?
- ☐ 0 – 1 year
- ☐ 1 – 3 years
- ☐ 3 – 6 years
- ☐ 6 – 10 years
- ☐ 10 – 20 years
- ☐ 20 – 40 years
- ☐ more than 40 years

Q5: How long do you work in your current job/position?
- ☐ 0 – 1 year
- ☐ 1 – 3 years
- ☐ 3 – 6 years
- ☐ 6 – 10 years
- ☐ 10 – 20 years
- ☐ 20 – 40 years
- ☐ more than 40 years

Q6: What is your current job/position?
- ☐ young detail designer
- ☐ senior product designer
- ☐ strategic product designer
- ☐ product manager
- ☐ development process manager
- ☐ other, please specify …………………………………………………………..

Q7: Do you use product data-based information in your daily tasks?
- ☐ not at all
- ☐ seldom
- ☐ regularly
- ☐ exclusively
- ☐ other, please specify …………………………………………………………..

Q8: In which country your current job is located?

- …

Q9: What continents you design products for?

- ☐ Africa
- ☐ Antarctica
- ☐ America
- ☐ Asia
- ☐ Australia
- ☐ Europe

Product/ tasks information

Q10: Which family of products you design, develop or manage?

- ☐ air conditioners
- ☐ cookers
- ☐ drying equipment
- ☐ cleaning appliances
- ☐ refrigeration equipment
- ☐ washing equipment
- ☐ other, please specify ……………………………………………………………

Q11: Which are the specific products you develop or manage? (Please specify all)

- …
- …
- …

Q12: What types of innovation does your company support? (Please specify all)

- …
- …
- …

Q13: What is the business strategy of your company?

- …

Knowing/ using big data

Q14: Do you use big data in your daily design tasks?

- ☐ yes
- ☐ no
- ☐ not relevant
- ☐ not known

Q15: From which data sources you obtain big data?

- ☐ sensors
- ☐ log files
- ☐ web resources (websites / forums)
- ☐ social media
- ☐ warehouse/repository
- ☐ product reports
- ☐ other, please specify ………………………………………………………………

Q16: How big is the product data you use in the design process?
- ☐ gigabytes
- ☐ terabytes
- ☐ petabytes
- ☐ exabytes
- ☐ zettabytes
- ☐ other, please specify ………………………………………………………………

Q17: What kind of data you work with?
- ☐ audio
- ☐ video
- ☐ textual
- ☐ numeric
- ☐ mixed
- ☐ other, please specify ………………………………………………………………

Q18: What is the nature of your raw (primary) data?
- ☐ structured
- ☐ multi-structured
- ☐ unstructured
- ☐ other, please specify ………………………………………………………………

Q19: Are your data time stamped big data?
- ☐ non-timed logically sequenced
- ☐ off-line batched
- ☐ on-line time sequenced
- ☐ on-line near real-time
- ☐ on-line real-time
- ☐ other, please specify ……………………………………………………………..

Q20: In which stage of the product life cycle your data is collected?
- ☐ beginning-of-life stage (when the product is designed and realized)
- ☐ middle-of-life stage (when the product is available on the market and used by the customer)
- ☐ end-of-life stage (when the product is dismissed or revamped)
- ☐ multiple stages concurrently
- ☐ other, please specify ………………………………………………………………

Q21: Are you satisfied with the contents of data you obtain/collect?

- □ If yes, please explain the reason:……………………………………………….
- □ If not, please explain the reason:……………………………………………..…

Q22: Are you satisfied with the quality of data you obtain/collect?

- □ If yes, please explain the reason:…………………………………………………
- □ If not, please explain the reason:……………………………………………..…..

Q23: What are the transformation steps that “the data” goes through before it is ready to be used in the design process?

- □ data sorting
- □ data cleaning
- □ data clustering
- □ data classification
- □ zettabytes
- □ other, please specify …………………………………………………………

Knowing/ using data analytics tools

Q24: Do you use data analytics software tools/packages in your product development tasks?

- □ yes
- □ no

If your answer is “no” , then what alternative techniques you use process data?

- □ manual processing
- □ mechanical processing
- □ other, please specify ……………………………………………………….

Q25: Which ones of the following data analytics tools do you use? *

- □ ADaMSoft
- □ Analytica
- □ BV4.1
- □ CLUTO
- □ COMSOL
- □ Dataiko
- □ DataMelt
- □ FreeMat
- □ GNU Octave
- □ JASP
- □ Knime
- □ Matlab
- □ MaxStat
- □ Microsoft Excel
- □ Microsoft SQL Server
- □ OpenStat
- □ Oracle

- □ PSPP
- □ R
- □ RapidMiner
- □ SAS
- □ Scilab
- □ Shogun
- □ SPSS Modeler
- □ SPSS Statistics
- □ Stata
- □ WEKA
- □ none of these tools
- □ other, please specify …………………………………………………………

If your answer to Q25 is ‘none’, what tools do you use to process your data? Please specify:..……………………………………………………………………………..

Q26: What data processing steps you use tools for?

- □ none
- □ data preparation and pre-processing (cleaning, structuring, clustering and classifying data)
- □ data processing (filtering, transformation, recognition and evaluation of patterns)
- □ data visualisation (representation of the primary and/or the processed data and patterns)
- □ others, please specify ………………………………………………………………

Q27: What are the data transformation steps that you cannot complete without data analytics tools?

- …
- …
- …

Q28: Do you use multiple data analytics tools to process your data?

- □ use one tool for all data transformation steps
- □ use different tools for different steps
- □ use multiple tools for one step

Q29: Who else is using the tools in your company?

- □ product designers
- □ data analysts
- □ operators
- □ others, please specify ………………………………………………………..

Q30: What difficulties you encounter using data analytics tools?

- □ complex user interface

- □ complex data processing
- □ complex programming
- □ others, please specify ………………………………………………………………..

Needed services / functions

Q31: How satisfied are you with data analytics tools you are using?

	Not satisfied									Very satisfied
– in terms of availability	1	2	3	4	5	6	7	8	9	10
– in terms of accessibility	1	2	3	4	5	6	7	8	9	10
– in terms of costs (price of the software)	1	2	3	4	5	6	7	8	9	10
– in terms of learning time	1	2	3	4	5	6	7	8	9	10
– in terms of mastering time	1	2	3	4	5	6	7	8	9	10
– in terms of usability (easy to use)	1	2	3	4	5	6	7	8	9	10
– in terms of functionalities	1	2	3	4	5	6	7	8	9	10
– in terms of time consumption	1	2	3	4	5	6	7	8	9	10
– in terms of efficiency (in processing data)	1	2	3	4	5	6	7	8	9	10
– in terms of flexibility (decision-making)	1	2	3	4	5	6	7	8	9	10
– in terms of adaptability (to different design tasks)	1	2	3	4	5	6	7	8	9	10

Q32: Are the data analytics tools you use easy to be mastered by designers?
- □ yes
- □ no, please justify ……………………………………………………………………

Q33: Do the outcomes of data analytics tools satisfy different design tasks?
- □ yes
- □ no

Q34: What are the things that you most dislike about data analytics tools you use?
- …
- …
- …
- …
- …

Q35: What are the things that you most like about data analytics tools you use?
- …
- …
- …

Q36: What functions of the tools you have never used?
- …
- …
- …

Q37: If a computer support would be to assist designers and execute all data analytics steps, how do you want it to be?

- …

Additional information/ Suggestions

Please in this last part of the questionnaire, add any remark, information, suggestion or recommendation that you judge helpful for the accomplishment of our study:

Remarks:

...……………………………………………………………………..............................

...……………………………………………………………………...................................

Additional Information:

...……………………………………………………………………........................

...……………………………………………………………………..................................

Suggestions and recommendations:

...………………………………………………………….

...……………………………………………………………………...................................

Q38: Are you satisfied with our questionnaire?

	Not satisfied									Very satisfied
□ in terms of utility	1	2	3	4	5	6	7	8	9	10
□ in terms of understanding	1	2	3	4	5	6	7	8	9	10
□ in terms of time consumption	1	2	3	4	5	6	7	8	9	10

Thank you for your participation to the questionnaire. Please wait for our future publication concerning the conceptualization of the data analytics toolbox for designers.

Appendix 2

Decomposition of the need theory (T_1)

Textual formulation of the theory:

- Data analytics tools have opened up a new path for generating knowledge for product enhancement.
- Product developers can achieve a perpetual enhancement of their products and services based on real life use, work and failure data.
- Numerous data analytics (software) tools and packages have been developed for extracting product-associated data, exploiting data analytics methods and tools in product enhancement.
- The proliferation of smart products forces companies to rethink and retool almost everything they do internally. Such products incorporate various self-learning, self-adaptation and self-management capabilities. They may actively generate, collect and communicate a large amount of data about their operational states and use circumstances, and can reason with these data.
- Most disliked things about the tools: Lack of needed functionalities to analyze data, Bad learning curve and customer support, slow learning time, and the absence of good training, learning program writing, and instruction information, Lack of ease of use caused by the heterogeneous user interfaces and the complexity of programming, Combination of qualitative and quantitative data is still a challenge, low performance for big databases as well as the non-liked proprietary format extension for files saving, Lack of adaptability to different design tasks and the complexity of interpreting the outcomes of the tools, Lack of desired variety of visualization and inadequacies of data display, High cost of software tools and packages, Unsolved bugs of the tools, Non intuitive, Dissatisfaction with the transferability of output data among tools, and the enormous amount of time involved in collecting relevant data, Getting confused by new releases of tools.
- Difficulties designers encounter using data analytics tools: complexity of the user interface, the complexity of programming, and the complexity of data processing.
- Designers expectations regarding a new data analytics computer support: Complete tool with high performances, Assisting user step by step, Advising user in his choices, Combining data from multiple sources, Providing a semantic support to data, Proposing multiple visualization options, Affordable and cheaper to get, Flexible in terms of tasks to allow, Intuitive tool (smart), Everywhere connected, Customizability of the tool, Accessibility to the tool at any time.
- Designer's needs: Step by step assistance, Advice at means selection, Multifold data visualization, Multi-channel data management, Blending datasets, Combining

qualitative and quantitative data, Permanent accessibility, Adaptation to user, Case-based reasoning, Learning from applications .

- We suggest that the expected data analytics solutions should be in harmony with the multiplicity and heterogeneity of data collection practices and analytical needs, and should be able to cope with incomplete data.
- Sophisticated data management functions that can be implemented as auxiliary functions of new toolboxes, to allow merging multiple data streams, facilitating data fusion, increasing computational performance, improving usability, and facilitating human interpretation.
- Smart semantic and procedural reasoning functions that use system intellect provided by artificial intelligence and system learning mechanisms, context information processing, situation awareness, strategy developments, and system adaptation capabilities. Ultimately, these are expected to support addressing all the needs and allowing the extraction of the meaning from MoL data, together with other lifecycle data.
- Our last proposal is that - having these novel affordances - developers and designers of white goods can optimally benefit from processing product, process and context data, and can generate innovative ideas for improvement of current product and for the creation of brand new products.

Legend:

$E_{1,x}$: refers to the derived entities
$A_{1,x}$: refers to the derived axioms from the textual theory.
$P^{D}_{1,x}$: refers to the derived postulates from the textual theory.
$P^{A}_{1,x}$: refers to the auxiliary postulates based on personal knowledge.
$(..)^{x} = E_{1,x}$; [..] = Relationship ; {..}= $\text{Proposition}_{1,x}$ (bloc x)

List of abbreviation:

Abbreviation	Designation	Abbreviation	Designation
BoL	Beginning-of-life	MoL	Middle-of-life
EoL	End-of-life	PLC	Product life cycle
AI	Artificial intelligence		

Extraction of entities:

Entity code	Denomination	Entity code	Denomination	Entity code	Denomination
$E_{1,1}$	Data analytics tool	$E_{1,2}$	Knowledge	$E_{1,3}$	Product
$E_{c,62}$	Designer	$E_{1,5}$	Use data	$E_{1,6}$	MoL data
$E_{1,7}$	MoL	$E_{1,8}$	PLC	$E_{1,9}$	Data
$E_{1,10}$	Work data	$E_{1,11}$	Failure data	$E_{1,12}$	Product-associated data

$E_{1,13}$	EoL	$E_{1,14}$	BoL	$E_{1,15}$	PLC data
$E_{1,16}$	EoL data	$E_{1,17}$	BoL data	$E_{1,18}$	Product developer
$E_{1,19}$	Developer	$E_{1,20}$	Data analytics package	$E_{1,21}$	Data analytics method
$E_{1,22}$	Product enhancement	$E_{1,23}$	Method	$E_{1,24}$	Smart product
$E_{1,25}$	Self-learning capability	$E_{1,26}$	Self-adaptive capability	$E_{1,27}$	Self-management capability
$E_{1,28}$	Capability	$E_{1,29}$	Operational state	$E_{1,30}$	State
$E_{1,31}$	Use circumstance	$E_{1,32}$	Circumstance	$E_{1,33}$	Learning curve
$E_{1,34}$	Curve	$E_{1,35}$	Time	$E_{1,36}$	Customer support
$E_{1,37}$	Support	$E_{1,38}$	Learning time	$E_{1,39}$	Good training
$E_{1,40}$	Training	$E_{1,41}$	Lack of ease of use	$E_{1,42}$	Heterogeous user interface
$E_{1,43}$	User interface	$E_{1,44}$	Complex programming	$E_{1,45}$	Programming
$E_{1,46}$	Instruction information	$E_{1,47}$	Information	$E_{1,48}$	Different design task
$E_{c,64}$	Design task	$E_{1,50}$	Complex outcome	$E_{1,51}$	Outcome
$E_{1,52}$	Difficult interpretation	$E_{c,72}$	Interpretation	$E_{1,54}$	Software tool
$E_{1,55}$	Computational tool	$E_{1,56}$	Software package	$E_{1,57}$	Tool
$E_{1,58}$	Toolbox	$E_{1,59}$	Unsolved bug	$E_{1,60}$	Bug
$E_{1,61}$	Relevant data	$E_{1,62}$	Interface	$E_{1,63}$	Data processing
$E_{1,64}$	Processing	$E_{1,65}$	Complete software tool	$E_{1,66}$	High performance
$E_{1,67}$	Performance	$E_{1,68}$	Affordable software tool	$E_{1,69}$	Step by step assistance
$E_{1,70}$	Guided assistance	$E_{1,71}$	Assistance	$E_{1,72}$	Multifold data visualization
$E_{1,73}$	Data visualization	$E_{1,74}$	Multi-channel data management	$E_{1,75}$	Data management
$E_{1,76}$	Blended datasets	$E_{1,77}$	Dataset	$E_{1,78}$	Combined data
$E_{1,79}$	Qualitative data	$E_{1,80}$	Quantitative data	$E_{1,81}$	Permanently accessible software tool
$E_{1,82}$	Customized software tool	$E_{1,83}$	Intuitive data analytics tool	$E_{1,84}$	Smart data analytics tool
$E_{1,85}$	Case-based reasoning	$E_{1,86}$	Reasoning	$E_{c,66}$	Semantic support
$E_{1,88}$	Data collection practice	$E_{1,89}$	Practice	$E_{1,90}$	Analytical need

$E_{1,91}$	Need	$E_{1,92}$	Sophisticated data management	$E_{1,93}$	Data stream
$E_{1,94}$	Data fusion	$E_{1,95}$	Computational performance	$E_{1,96}$	Fusion
$E_{1,97}$	Usability	$E_{1,98}$	Human interpretation	$E_{1,99}$	Smart semantics
$E_{1,100}$	Semantics	$E_{1,101}$	Procedural reasoning	$E_{1,102}$	MoL data meaning
$E_{1,103}$	System intellect	$E_{1,104}$	AI	$E_{1,105}$	System learning mechanism
$E_{1,106}$	Context information processing	$E_{1,107}$	Situation awareness	$E_{1,108}$	Strategy development
$E_{1,109}$	System adaptation capability	$E_{1,110}$	PLC data meaning	$E_{1,111}$	Data meaning
$E_{1,112}$	Meaning	$E_{1,113}$	Intelligence	$E_{1,114}$	Learning mechanism
$E_{1,115}$	Mechanism	$E_{c,68}$	Awareness	$E_{1,117}$	Development
$E_{1,118}$	Choice	$E_{1,119}$	Application	$E_{1,120}$	Data analytics solution
$E_{1,121}$	Solution	$E_{c,78}$	New product	$E_{1,123}$	Incomplete data
$E_{1,124}$	Knowledge representation	$E_{1,125}$	Representation	$E_{1,126}$	Learning
$E_{1,127}$`	Natural language processing	$E_{1,128}$	Machine learning	$E_{c,80}$	Task

Axiomatization of the need theory (T1):

$A_{1,1}$: (Data analytics tools)[1] [generate] (knowledge)[2]
$A_{1,2}$: (Knowledge)[2] [is generated for] (product enhancement)[22]
$A_{1,3}$: (Designers)[c,62] [enhance] (products)[3]
$A_{1,4}$: (Designers)[c,62] [are] (product developers)[18]
$A_{1,5}$: (Product enhancement)[22] [is based on] (use data)[5]
$A_{1,6}$: (Product enhancement)[22] [is based on] (work data)[10]
$A_{1,7}$: (Product enhancement)[22] [is based on] (failure data)[11]
$A_{1,8}$: (Data analytics tools)[1] [extract] (product-associated data)[12]
$A_{1,9}$: (Data analytics packages)[20] [extract] (product-associated data)[12]
$A_{1,10}$: (Data analytics tools)[1] [exploit] (data analytics methods)[21]
$A_{1,11}$: (Data analytics packages)[20] [exploit] (data analytics methods)[21]
$A_{1,12}$: (Data analytics methods)[21] [are used for] (product enhancement)[22]
$A_{1,13}$: (Smart products)[24] [incorporate] (self-learning capabilities)[25]
$A_{1,14}$: (Smart products)[24] [incorporate] (self-adaption capabilities)[26]
$A_{1,15}$: (Smart products)[24] [incorporate] (self-management capabilities)[27]
$A_{1,16}$: (Smart products)[24] [collect their] (operational state)[29]
$A_{1,17}$: (Smart products)[24] [communicate their] (operational state)[29]
$A_{1,18}$: (Smart products)[24] [reason with their] (operational state)[29]
$A_{1,19}$: (Smart products)[24] [collect their] (use circumstances)[31]

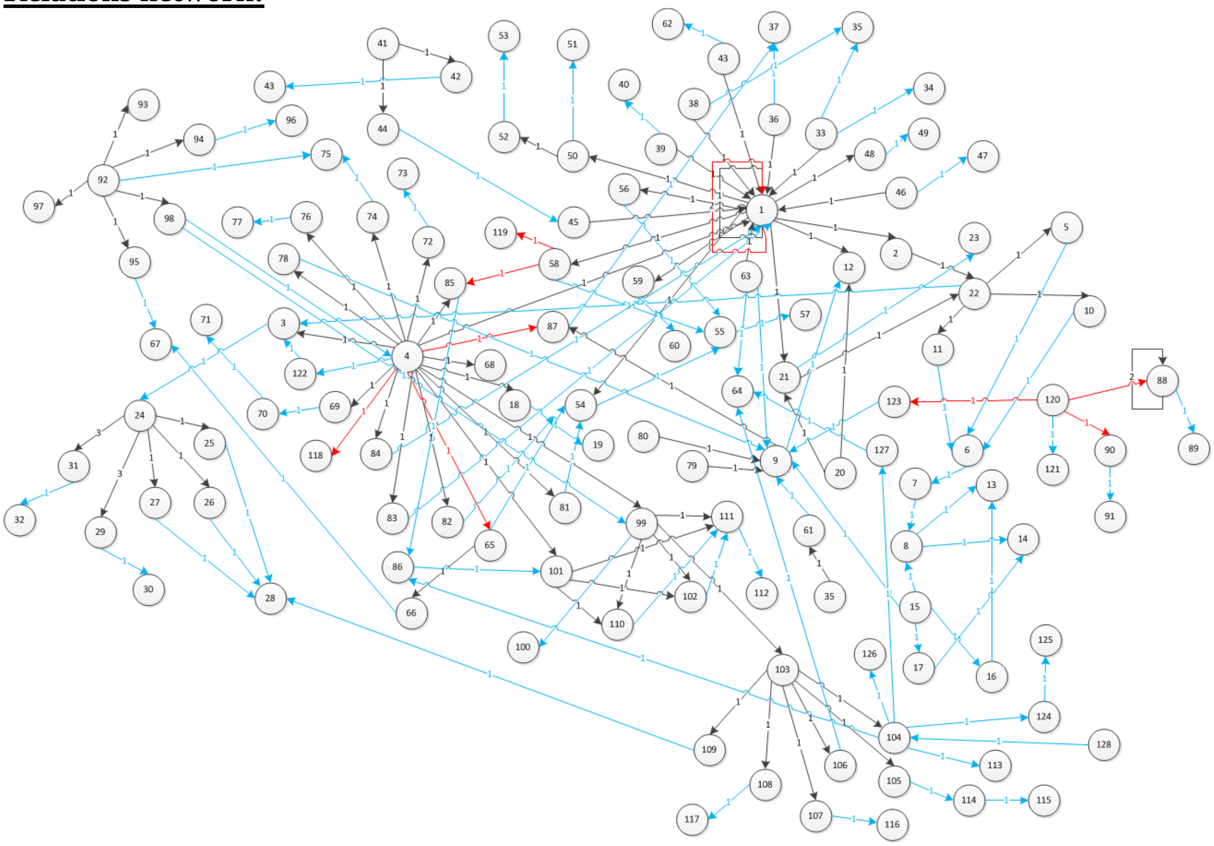
$A_{1,20}$: (Smart products)[24] [communicate their] (use circumstances)[31]
$A_{1,21}$: (Smart products)[24] [reason with their] (use circumstances)[31]
$A_{1,22}$: (Learning curve)[33] of (data analytics tools)[1] [is bad]
$A_{1,23}$: (Customer support)[36] of (data analytics tools)[1] [is bad]
$A_{1,24}$: (Learning time)[38] of (data analytics tools)[1] [is slow]
$A_{1,25}$: (Good training)[39] [is absent for] (data analytics tools)[1]
$A_{1,26}$: (Lack of ease of use)[41] [is caused by] (heterogeneous user interfaces)[42]
$A_{1,27}$: (Lack of ease of use)[41] [is caused by] (complex programming)[44]
$A_{1,29}$: (Data analytics tools)[1] [are not adaptable to] (different design tasks)[48]
$A_{1,30}$: (Data analytics tools)[1] [provide] (complex outcomes)[50]
$A_{1,31}$: (Complex outcomes)[50] [cause] (difficult interpretation)[52]
$A_{1,32}$: (Data analytics tools)[1] [include] (software tools)[54]
$A_{1,33}$: (Data analytics tools)[1] [include] (software packages)[56]
$A_{1,34}$: (Data analytics tools)[1] [include] (toolboxes)[58]
$A_{1,35}$: (Data analytics tools)[1] [contain] (unsolved bugs)[59]
$A_{1,36}$: (Data analytics tools)[1] [are not intuitive]
$A_{1,37}$: (Time)[35] (is consumed in collecting) (relevant data)[61]
$A_{1,38}$: (User interface)[43] of (data analytics tools)[1] [is complex]
$A_{1,39}$: (Programming)[45] within (data analytics tools)[1] [is complex]
$A_{1,40}$: (Data processing)[63] within (data analytics tools)[1] [is complex]
$A_{1,41}$: (A complete software tool)[65] [has] (high performances)[66]
$A_{1,42}$: (Designers)[c,62] [need an] (affordable software tool)[68]
$A_{1,43}$: (Designers)[c,62] [need] (step by step assistance)[69]
$A_{1,44}$: (Designers)[c,62] [need] (multifold data visualization)[72]
$A_{1,45}$: (Designers)[c,62] [need] (multi-channel data management)[74]
$A_{1,46}$: (Designers)[c,62] [need] (blended datasets)[76]
$A_{1,47}$: (Designers)[c,62] [need] (combined data)[78]
$A_{1,48}$: (Qualitative data)[79] [is included in] (data)[9]
$A_{1,49}$: (Quantitative data)[80] [is included in] (data)[9]
$A_{1,50}$: (Designers)[c,62] [need a] (permanently accessible software tool)[81]
$A_{1,51}$: (Designers)[c,62] [need a] (customized software tool)[82]
$A_{1,52}$: (Designers)[c,62] [need an] (intuitive data analytics tool)[83]
$A_{1,53}$: (Designers)[c,62] [need] (smart data analytics tool)[84]
$A_{1,54}$: (Designers)[c,62] [need] (case-based reasoning)[85]
$A_{1,55}$: (Semantic support)[c,66] [is provided to] (data)[9]
$A_{1,56}$: (Data collection practices)[88] [are multiple]
$A_{1,57}$: (Data collection practices)[88] [are heterogeneous]
$A_{1,58}$: (Sophisticated data management)[92] [merges] (data streams)[93]
$A_{1,59}$: (Sophisticated data management)[92] [facilitates] (data fusion)[94]
$A_{1,60}$: (Sophisticated data management)[92] [increases] (computational performances)[95]
$A_{1,61}$: (Sophisticated data management)[92] [improves] (usability)[97]
$A_{1,62}$: (Sophisticated data management)[92] [facilitates] (human interpretation)[98]
$A_{1,63}$: (Designers)[c,62] [need] (smart semantics)[99]
$A_{1,64}$: (Designers)[c,62] [need] (procedural reasoning)[101]
$A_{1,65}$: (Smart semantics)[99] [extracts] (MoL data meaning)[102]
$A_{1,66}$: (Smart semantics)[99] [extracts] (PLC data meaning)[110]

$A_{1,67}$: (Procedural reasoning)[101] [extracts] (MoL data meaning)[102]
$A_{1,68}$: (Procedural reasoning)[101] [extracts] (PLC data meaning)[110]
$A_{1,69}$: (Smart semantics)[99] [uses] (system intellect)[103]
$A_{1,70}$: (System intellect)[103] [is provided by] (AI)[104]
$A_{1,71}$: (System intellect)[103] [is provided by] (system learning mechanisms)[105]
$A_{1,72}$: (System intellect)[103] [is provided by] (context information processing)[106]
$A_{1,73}$: (System intellect)[103] [is provided by] (situation awareness)[107]
$A_{1,74}$: (System intellect)[103] [is provided by] (strategy development)[108]
$A_{1,75}$: (System intellect)[103] [is provided by] (system adaptation capabilities)[109]
$A_{1,76}$: (Smart semantics)[99] [extracts] (data meaning)[111]
$A_{1,77}$: (Procedural reasoning)[101] [extracts] (data meaning)[111]
$A_{1,78}$: (Designers)[c,62] [use] (data analytics tools)[1]
$P^D_{1,1}$: (Designers)[c,62] [want to have a] (complete software tool)[65]
$P^D_{1,2}$: (Designers)[c,62] [want to be advised in their] (choices)[118]
$P^D_{1,3}$: (Toolbox)[58] [should learn from its] (application)[119]
$P^D_{1,4}$: (Toolbox)[58] [should include] (case-based reasoning)[85]
$P^D_{1,5}$: (Data analytics tool)[1] [should be intuitive]
$P^D_{1,6}$: (Data analytics tool)[1] [should be smart]
$P^D_{1,7}$: (Designers)[c,62] [want to have] (semantic support)[c,66]
$P^D_{1,8}$: (Data analytics solutions)[120] [should be in harmony with] (data collection practices)[88]
$P^D_{1,9}$: (Data analytics solutions)[120] [should be in harmony with] (analytical needs)[90]
$P^D_{1,10}$: (Data analytics solutions)[120] [should cope with] (incomplete data)[123]
$P^A_{1,1}$: (Use data)[5] [belong to] (MoL data)[6]
$P^A_{1,2}$: (MoL data)[6] [are collected during] (MoL)[7]
$P^A_{1,3}$: (MoL)[7] [belongs to] (PLC)[8]
$P^A_{1,4}$: (PLC)[8] [includes] (EoL)[13]
$P^A_{1,5}$: (PLC)[8] [includes] (BoL)[14]
$P^A_{1,6}$: (PLC data)[15] [include] (EoL data)[16]
$P^A_{1,7}$: (PLC data)[15] [include] (BoL data)[17]
$P^A_{1,8}$: (PLC data)[15] [is collected during] (PLC)[8]
$P^A_{1,9}$: (EoL data)[16] [is collected during] (EoL)[13]
$P^A_{1,10}$: (BoL data)[17] [is collected during] (BoL)[14]
$P^A_{1,11}$: (PLC data)[15] [are] (data)[9]
$P^A_{1,12}$: (Work data)[10] [belongs to] (MoL data)[6]
$P^A_{1,13}$: (Failure data)[11] [belongs to] (MoL data)[6]
$P^A_{1,14}$: (Product developer)[18] [is a] (developer)[19]
$P^A_{1,15}$: (Product enhancement)[22] [concerns] (products)[3]
$P^A_{1,16}$: (Data analytics method)[21] [is a] (method)[23]
$P^A_{1,17}$: (Smart product)[24] [is a] (product)[3]
$P^A_{1,18}$: (Self-learning capability)[25] [is a] (capability)[28]
$P^A_{1,19}$: (Sel-adaptation capability)[26] [is a] (capability)[28]
$P^A_{1,20}$: (Self-management capability)[27] [is a] (capability)[28]
$P^A_{1,21}$: (Operational state)[29] [is a] (state)[30]
$P^A_{1,22}$: (Use circumstance)[31] [is a] (circumstance)[32]
$P^A_{1,23}$: (Learning curve)[33] [is a] (curve)[34]

$P^{A}_{1,24}$: (Learning curve)[33] [is proportional with] the (time)[35]
$P^{A}_{1,25}$: (Customer support)[36] [is a] (support)[37]
$P^{A}_{1,26}$: (Learning time)[38] [is a] (time)[35]
$P^{A}_{1,27}$: (Good training)[39] [is a] (training)[40]
$P^{A}_{1,28}$: (Heterogeneous user interface)[42] [is a] (user interface)[43]
$P^{A}_{1,29}$: (Complex programming)[44] [is a] (programming)[45]
$P^{A}_{1,30}$: (Instruction information)[46] [is an] (information)[47]
$P^{A}_{1,31}$: (Different design tasks)[48] [are] (design tasks)[c,64]
$P^{A}_{1,32}$: (Complex outcomes)[50] [are] (outcomes)[51]
$P^{A}_{1,33}$: (Difficult interpretation)[52] [is an] (interpretation)[c,72]
$P^{A}_{1,34}$: (Software tools)[54] [belong to] (computational tools)[55]
$P^{A}_{1,35}$: (Software packages)[56] [belong to] (computational tools)[55]
$P^{A}_{1,36}$: (Computational tools)[55] [are] (tools)[57]
$P^{A}_{1,37}$: (Toolboxes)[58] [belong to] (computational tools)[55]
$P^{A}_{1,38}$: (Unsolved bug)[59] [is a] (bug)[60]
$P^{A}_{1,39}$: (Relevant data)[61] [are] (data)[9]
$P^{A}_{1,40}$: (User interface)[43] [in an] (interface)[62]
$P^{A}_{1,41}$: (Data processing)[63] [is a] (processing)[64]
$P^{A}_{1,42}$: (Complete software tool)[65] [is a] (software tool)[54]
$P^{A}_{1,43}$: (High performance)[66] [is a] (performance)[67]
$P^{A}_{1,44}$: (Affordable software tool)[68] [is a] (software tool)[54]
$P^{A}_{1,45}$: (Step by step assistance)[69] [is a] (guided assistance)[70]
$P^{A}_{1,46}$: (Guided assistance)[70] [is an] (assistance)[71]
$P^{A}_{1,47}$: (Multifold data visualization)[72] [is a] (data visualization)[73]
$P^{A}_{1,48}$: (Multi-channel data management)[74] [belongs to] (data management)[75]
$P^{A}_{1,49}$: (Blended datasets)[76] [are] (datasets)[77]
$P^{A}_{1,50}$: (Combined data)[78] [group] (data)[9]
$P^{A}_{1,51}$: (Permanently accessible software tool)[81] [is a] (software tool)[54]
$P^{A}_{1,52}$: (Customized software tool)[82] [is a] (software tool)[54]
$P^{A}_{1,53}$: (Intuitive data analytics tool)[83] [belongs to] (data analytics tools)[1]
$P^{A}_{1,54}$: (Smart data analytics tool)[84] [belongs to] (data analytics tools)[1]
$P^{A}_{1,55}$: (Case-based reasoning)[85] [is a] (reasoning)[86]
$P^{A}_{1,56}$: (Semantic support)[c,66] [is a] (support)[37]
$P^{A}_{1,57}$: (Data collection practice)[88] [is a] (practice)[89]
$P^{A}_{1,58}$: (Analytical need)[90] [is a] (need)[91]
$P^{A}_{1,59}$: (Incomplete data)[123] [belong to] (data)[9]
$P^{A}_{1,60}$: (Sophisticated data management)[92] [belong to] (data management)[75]
$P^{A}_{1,61}$: (Data fusion)[94] [is a] (fusion)[96]
$P^{A}_{1,62}$: (Computational performance)[95] [is a] (performance)[67]
$P^{A}_{1,63}$: (Human interpretation)[98] [is an] (interpretation)[c,72]
$P^{A}_{1,64}$: (Human interpretation)[98] [is done by] (designers)[c,62]
$P^{A}_{1,65}$: (Smart semantics)[99] [are] (semantics)[100]
$P^{A}_{1,66}$: (Procedural reasoning)[101] [is a] (reasoning)[86]
$P^{A}_{1,67}$: (MoL data meaning)[102] [is a] (data meaning)[111]
$P^{A}_{1,68}$: (PLC data meaning)[110] [is a] (data meaning)[111]
$P^{A}_{1,69}$: (Data meaning)[111] [is a] (meaning)[112]

$P^{A}_{1,70}$: (AI)^{104} [is an] (intelligence)^{113}
$P^{A}_{1,71}$: (System learning mechanism)^{105} [is a] (learning mechanism)^{114}
$P^{A}_{1,72}$: (Learning mechanism)^{114} [is a] (mechanism)^{115}
$P^{A}_{1,73}$: (Context information processing)^{106} [is] (processing)^{64}
$P^{A}_{1,74}$: (Situation awareness)^{107} [is an] (awareness)^{c,68}
$P^{A}_{1,75}$: (Strategy development)^{108} [is a] (development)^{117}
$P^{A}_{1,76}$: (System adaptation capability)^{109} [is a] (capability)^{28}
$P^{A}_{1,77}$: (Data analytics solutions)^{120} [are] (solutions)^{121}
$P^{A}_{1,78}$: (Designers)^{c,62} [create] (new products)^{c,78}
$P^{A}_{1,79}$: (New product)^{c,78} [is a] (product)^{3}
$P^{A}_{1,80}$: (AI)^{104} [includes] (reasoning)^{86}
$P^{A}_{1,81}$: (AI)^{104} [includes] (knowledge representation)^{124}
$P^{A}_{1,82}$: (Knowledge representation)^{124} [is a] (representation)^{125}
$P^{A}_{1,83}$: (AI)^{104} [includes] (learning)^{126}
$P^{A}_{1,84}$: (AI)^{104} [includes] (natural language processing)^{127}
$P^{A}_{1,85}$: (Natural language processing)^{127} [is a] (processing)^{64}
$P^{A}_{1,86}$: (Machine learning)^{128} [belongs to] (AI)^{104}
$P^{A}_{1,87}$: (Data processing)^{63} [processes] (data)^{9}
$P^{A}_{1,88}$: (Product-accociated data)^{12} [is a] (data)^{9}
$P^{A}_{1,89}$: (Design task)^{c,64} [is a] (task)^{c,80}

Relations network:



The numbers included in the circles represent the number of relationships between entities.

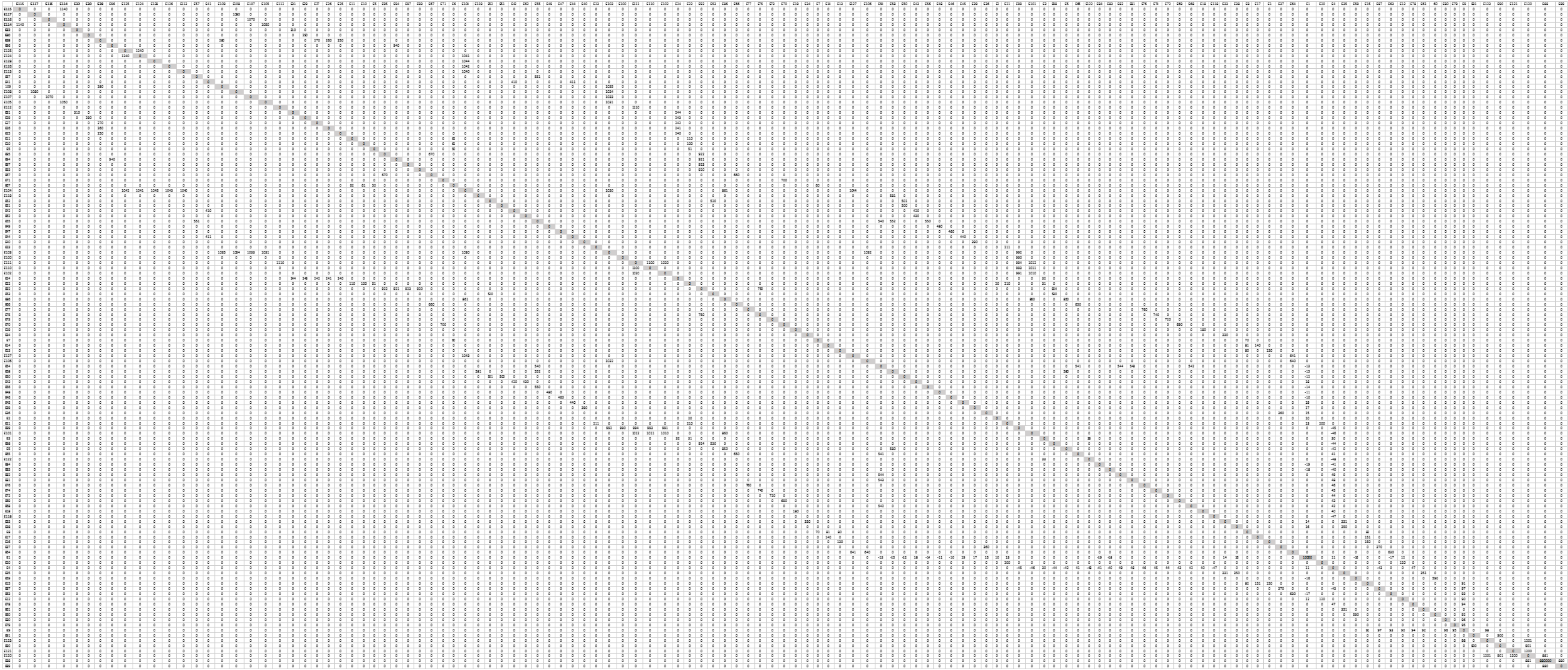
Matrix decomposition: Original matrix

Matrix coding:

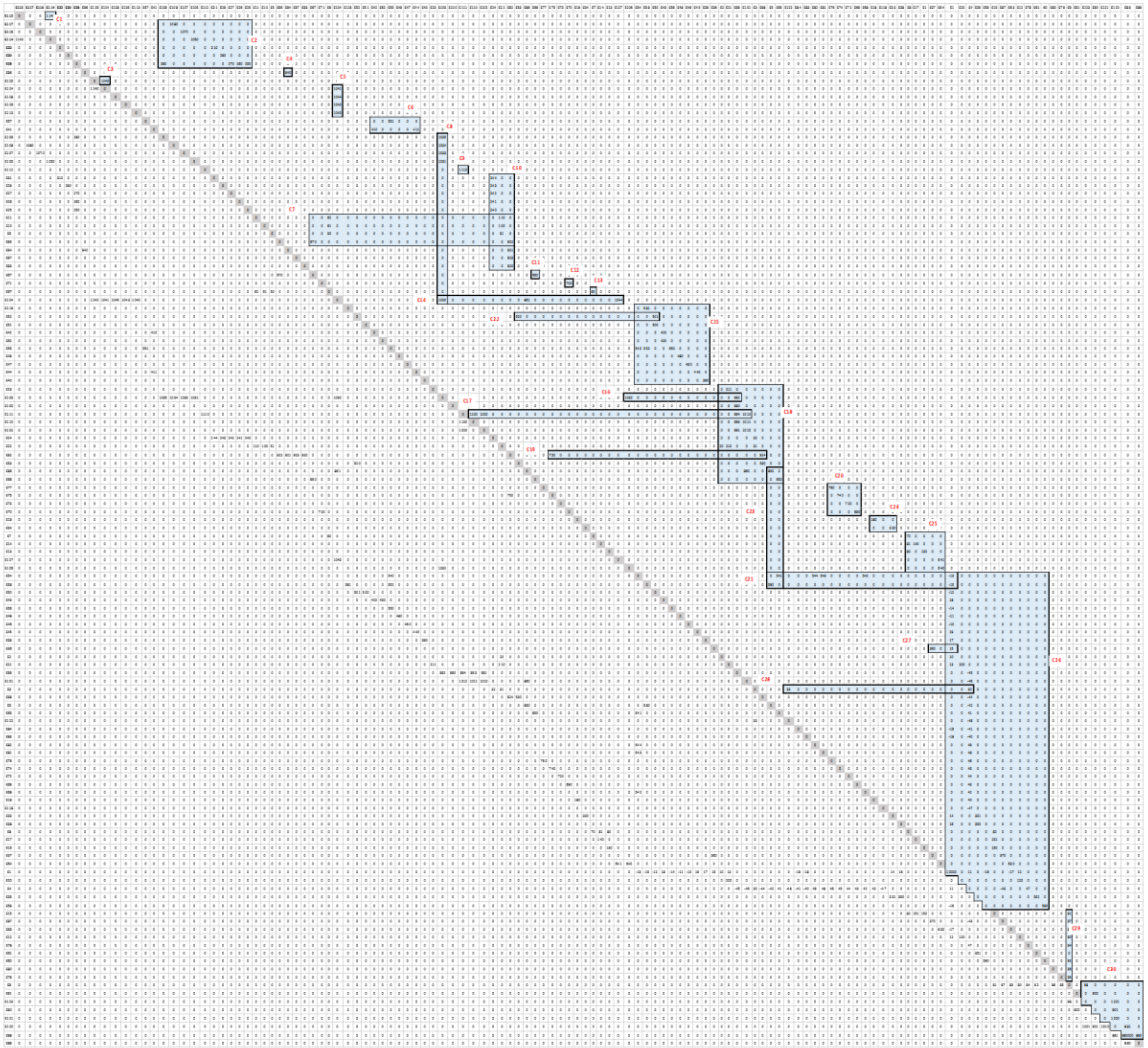
Code	Relationships	Code	Relationships	Code	Relationships	Code	Relationships
10	$A_{1,1}$	11	$A_{1,78}$	12	$A_{1,8}$	13	$A_{1,10}$
14	$A_{1,22}$	15	$A_{1,23}$	16	$A_{1,24}$	17	$A_{1,25}$
18	$A_{1,38}$	19	$A_{1,39}$	-10	$A_{1,28}$	-11	$A_{1,29}$
-12	$A_{1,30}$	-13	$A_{1,32}$	-14	$A_{1,33}$	-15	$A_{1,34}$
-16	$A_{1,35}$	-17	$A_{1,40}$	-18	$P^A_{1,53}$	-19	$P^A_{1,54}$
20	$A_{1,2}$	30	$A_{1,3}$	31	$P^A_{1,15}$	32	$P^A_{1,17}$
33	$P^A_{1,79}$	40	$A_{1,4}$	41	$P^D_{1,1}$	42	$A_{1,42}$
43	$A_{1,43}$	44	$A_{1,44}$	45	$A_{1,45}$	46	$A_{1,46}$
47	$A_{1,47}$	48	$A_{1,50}$	49	$A_{1,51}$	-40	$A_{1,52}$
-41	$A_{1,53}$	-42	$A_{1,54}$	-43	$P^D_{1,7}$	-44	$P^A_{1,67}$
-45	$A_{1,63}$	-46	$A_{1,64}$	-47	$P^D_{1,2}$	-48	$P^A_{1,78}$
50	$P^A_{1,1}$	51	$A_{1,5}$	60	$P^A_{1,2}$	61	$P^A_{1,12}$
62	$P^A_{1,13}$	70	$P^A_{1,3}$	80	$P^A_{1,4}$	81	$P^A_{1,5}$
82	$P^A_{1,8}$	90	$P^A_{1,88}$	91	$P^A_{1,11}$	92	$P^A_{1,39}$
93	$P^A_{1,87}$	94	$P^A_{1,50}$	95	$A_{1,48}$	96	$A_{1,49}$
97	$A_{1,55}$	98	$P^A_{1,59}$	100	$A_{1,6}$	110	$A_{1,7}$
120	$A_{1,9}$	130	$P^A_{1,9}$	140	$P^A_{1,10}$	150	$P^A_{1,6}$
151	$P^A_{1,7}$	180	$P^A_{1,14}$	200	$A_{1,11}$	210	$A_{1,12}$
211	$P^A_{1,16}$	240	$A_{1,13}$	241	$A_{1,14}$	242	$A_{1,15}$
243	$A_{1,16}$; $A_{1,17}$; $A_{1,18}$	244	$A_{1,19}$; $A_{1,20}$; $A_{1,21}$	250	$P^A_{1,18}$	260	$P^A_{1,19}$
270	$P^A_{1,20}$	280	$P^A_{1,76}$	290	$P^A_{1,21}$	310	$P^A_{1,22}$
330	$P^A_{1,23}$	331	$P^A_{1,24}$	350	$P^A_{1,26}$	351	$A_{1,37}$
360	$P^A_{1,25}$	370	$P^A_{1,56}$	390	$P^A_{1,27}$	410	$A_{1,26}$
411	$A_{1,27}$	420	$P^A_{1,28}$	430	$P^A_{1,40}$	440	$P^A_{1,29}$
460	$P^A_{1,30}$	480	$P^A_{1,31}$	500	$P^A_{1,32}$	501	$A_{1,31}$
520	$P^A_{1,33}$	530	$P^A_{1,63}$	540	$P^A_{1,34}$	541	$P^A_{1,42}$
542	$P^A_{1,44}$	543	$P^A_{1,51}$	544	$P^A_{1,52}$	550	$P^A_{1,35}$
551	$P^A_{1,36}$	552	$P^A_{1,37}$	580	$P^D_{1,4}$	581	$P^D_{1,3}$
590	$P^A_{1,38}$	630	$P^A_{1,41}$	640	$P^A_{1,73}$	641	$P^A_{1,85}$
650	$A_{1,41}$	660	$P^A_{1,43}$	670	$P^A_{1,62}$	690	$P^A_{1,45}$
700	$P^A_{1,46}$	720	$P^A_{1,47}$	740	$P^A_{1,48}$	750	$P^A_{1,60}$
760	$P^A_{1,49}$	850	$P^A_{1,55}$	860	$P^A_{1,66}$	861	$P^A_{1,80}$
880	$P^A_{1,57}$	881	$P^D_{1,8}$	900	$P^A_{1,58}$	901	$P^D_{1,9}$
920	$A_{1,58}$	921	$A_{1,59}$	922	$A_{1,60}$	923	$A_{1,61}$
924	$A_{1,62}$	940	$P^A_{1,61}$	990	$P^A_{1,65}$	991	$A_{1,65}$
992	$A_{1,69}$	993	$A_{1,66}$	994	$A_{1,76}$	1010	$A_{1,67}$
1011	$A_{1,68}$	1012	$A_{1,77}$	1020	$P^A_{1,67}$	1030	$A_{1,70}$
1031	$A_{1,71}$	1032	$A_{1,72}$	1033	$A_{1,73}$	1034	$A_{1,74}$
1035	$A_{1,75}$	1040	$P^A_{1,70}$	1041	$P^A_{1,81}$	1042	$P^A_{1,83}$
1043	$P^A_{1,84}$	1044	$P^A_{1,86}$	1050	$P^A_{1,71}$	1070	$P^A_{1,74}$
1080	$P^A_{1,75}$	1100	$P^A_{1,68}$	1110	$P^A_{1,69}$	1140	$P^A_{1,72}$
1200	$P^A_{1,77}$	1201	$P^D_{1,10}$	1240	$P^A_{1,82}$	10000	$A_{1,36}$; $P^D_{1,5}$; $P^D_{1,6}$
880000	$A_{1,56}$; $A_{1,57}$						

Matrix after coding:

Matrix partitioning using Matlab:



Clusters representation:



Projection:

Cluster 1:

	E114
E115	1140

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,72}$: (Learning mechanism)^{114} [is a] (mechanism)^{115}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 2:

	E109	E108	E107	E105	E112	E31	E29	E27	E26	E25
E117	0	1080	0	0	0	0	0	0	0	0
E116	0	0	1070	0	0	0	0	0	0	0
E114	0	0	0	1050	0	0	0	0	0	0
E32	0	0	0	0	0	310	0	0	0	0
E30	0	0	0	0	0	0	290	0	0	0
E28	280	0	0	0	0	0	0	270	260	250

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,76}$: (System adaptation capability)^{109} [is a] (capability)^{28}
$P^{A}_{1,75}$: (Strategy development)^{108} [is a] (development)^{117}
$P^{A}_{1,74}$: (Situation awareness)^{107} [is an] (awareness)^{116}
$P^{A}_{1,71}$: (System learning mechanism)^{105} [is a] (learning mechanism)^{114}
$P^{A}_{1,22}$: (Use circumstance)^{31} [is a] (circumstance)^{32}
$P^{A}_{1,21}$: (Operational state)^{29} [is a] (state)^{30}
$P^{A}_{1,20}$: (Self-management capability)^{27} [is a] (capability)^{28}
$P^{A}_{1,19}$: (Sel-adaptation capability)^{26} [is a] (capability)^{28}
$P^{A}_{1,18}$: (Self-learning capability)^{25} [is a] (capability)^{28}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 3:

	E124
E125	1240

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,82}$: (Knowledge representation)^{124} [is a] (representation)^{125}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 4:

	E94
E96	940

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,61}$: (Data fusion)[94] [is a] (fusion)[96]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 5:

	E104
E124	1041
E128	1044
E126	1042
E113	1040

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,81}$: (AI)[104] [includes] (knowledge representation)[124]
$P^{A}_{1,86}$: (Machine learning)[128] [belongs to] (AI)[104]
$P^{A}_{1,83}$: (AI)[104] [includes] (learning)[126]
$P^{A}_{1,70}$: (AI)[104] [is an] (intelligence)[113]

Derived propositions from the blocks:

Proposition$_{1,1}$ ($P^{A}_{1,81}$; $P^{A}_{1,86}$; $P^{A}_{1,83}$; $P^{A}_{1,70}$) : {AI is intelligence that includes learning, machine learning and knowledge representation}.

Cluster 6:

	E42	E62	E55	E49	E47	E44
E57	0	0	551	0	0	0
E41	410	0	0	0	0	411

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,36}$: (Computational tools)[55] [are] (tools)[57]
$A_{1,26}$: (Lack of ease of use)[41] [is caused by] (heterogeneous user interfaces)[42]
$A_{1,27}$: (Lack of ease of use)[41] [is caused by] (complex programming)[44]

Derived propositions from the blocks:
Proposition$_{1,2}$($A_{1,26}$; $A_{1,27}$) : {Lack of ease of use is caused by heterogenous user interfaces and complex programming}.

Cluster 7:

	E67	E71	E6	E104	...	E24	E22	E92
E11	0	0	62	0	...	0	110	0
E10	0	0	61	0	...	0	100	0
E5	0	0	50	0	...	0	51	0
E95	670	0	0	0	...	0	0	922

Axioms and postulates mentioned in this cluster are:

$P^A_{1,62}$: (Computational performance)[95] [is a] (performance)[67]
$P^A_{1,13}$: (Failure data)[11] [belongs to] (MoL data)[6]
$P^A_{1,12}$: (Work data)[10] [belongs to] (MoL data)[6]
$P^A_{1,1}$: (Use data)[5] [belongs to] (MoL data)[6]
$A_{1,7}$: (Product enhancement)[22] [is based on] (failure data)[11]
$A_{1,6}$: (Product enhancement)[22] [is based on] (work data)[10]
$A_{1,5}$: (Product enhancement)[22] [is based on] (use data)[5]
$A_{1,60}$: (Sophisticated data management)[92] [increases] (computational performances)[95]

Derived propositions from the blocks:

Proposition$_{1,3}$($P^A_{1,13}$; $P^A_{1,12}$;$P^A_{1,1}$; $A_{1,7}$; $A_{1,6}$; $A_{1,5}$) : {Product enhancement is based on failure, work and use data that belong to MoL data}.

Cluster 8:

	E103
E109	1035
E108	1034
E107	1033
E105	1031
E112	0
...	...
E67	0
E104	1030

Axioms and postulates mentioned in this cluster are:

$A_{1,75}$: (System intellect)[103] [is provided by] (system adaptation capabilities)[109]
$A_{1,74}$: (System intellect)[103] [is provided by] (strategy development)[108]
$A_{1,73}$: (System intellect)[103] [is provided by] (situation awareness)[107]
$A_{1,71}$: (System intellect)[103] [is provided by] (system learning mechanisms)[105]

$A_{1,70}$: (System intellect)[103] [is provided by] (AI)[104]

Derived propositions from the blocks:

Proposition$_{1,4}$($A_{1,75}$; $A_{1,74}$; $A_{1,73}$; $A_{1,71}$; $A_{1,70}$) : {System intellect is provided by (i) system adaptation capabilities, (ii)strategy development, (iii) situation awarness, (iv) system learning mechanisms, and (v) AI}.

Cluster 9:

	E111
E112	1110

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,69}$: (Data meaning)[111] [is a] (meaning)[112]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 10:

	E24	E22	E92
E31	244	0	0
E29	243	0	0
E27	242	0	0
E26	241	0	0
E25	240	0	0
E11	0	110	0
E10	0	100	0
E5	0	51	0
E95	0	0	922
E94	0	0	921
E97	0	0	923
E93	0	0	920

Axioms and postulates mentioned in this cluster are:

$A_{1,19}$: (Smart products)[24] [collect their] (use circumstances)[31]
$A_{1,20}$: (Smart products)[24] [communicate their] (use circumstances)[31]
$A_{1,21}$: (Smart products)[24] [reason with their] (use circumstances)[31]
$A_{1,16}$: (Smart products)[24] [collect their] (operational state)[29]
$A_{1,17}$: (Smart products)[24] [communicate their] (operational state)[29]
$A_{1,18}$: (Smart products)[24] [reason with their] (operational state)[29]
$A_{1,15}$: (Smart products)[24] [incorporate] (self-management capabilities)[27]
$A_{1,14}$: (Smart products)[24] [incorporate] (self-adaption capabilities)[26]
$A_{1,13}$: (Smart products)[24] [incorporate] (self-learning capabilities)[25]

$A_{1,7}$: (Product enhancement)[22] [is based on] (failure data)[11]
$A_{1,6}$: (Product enhancement)[22] [is based on] (work data)[10]
$A_{1,5}$: (Product enhancement)[22] [is based on] (use data)[5]

$A_{1,60}$: (Sophisticated data management)[92] [increases] (computational performances)[95]
$A_{1,59}$: (Sophisticated data management)[92] [facilitates] (data fusion)[94]
$A_{1,61}$: (Sophisticated data management)[92] [improves] (usability)[97]
$A_{1,58}$: (Sophisticated data management)[92] [merges] (data streams)[93]

Derived propositions from the blocks:

Proposition$_{1,5}$($A_{1,19}$; $A_{1,20}$; $A_{1,21}$; $A_{1,16}$; $A_{1,17}$; $A_{1,18}$; $A_{1,15}$; $A_{1,14}$; $A_{1,13}$) : {Smart products incorporate self-management, self-adaptation and self-learning capabilities and collect, communicate and reason with their use-circumstances and operational state}.

Proposition$_{1,6}$($A_{1,7}$; $A_{1,6}$; $A_{1,5}$) : {Product enhancement is based on failure, work and use data.

Proposition$_{1,7}$($A_{1,60}$; $A_{1,59}$; $A_{1,61}$; $A_{1,58}$) : Sophisticated data management increases computational performance, facilitates data fusion, improves usability and merges data streams}.

Cluster 11:

	E66
E67	660

Axioms and postulates mentioned in this cluster are:

$P^A_{1,43}$: (High performance)[66] [is a] (performance)[67]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 12:

	E70
E71	700

Axioms and postulates mentioned in this cluster are:

$P^A_{1,46}$: (Guided assistance)[70] [is an] (assistance)[71]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 13:

	E7
E67	60

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,2}$: (MoL data)[6] [are collected during] (MoL)[7]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 14:

	E103	E100	...	E92	E86	E66	...	E13	E127
E104	1030	0	...	0	861	0	...	0	1044

Axioms and postulates mentioned in this cluster are:

$A_{1,70}$: (System intellect)[103] [is provided by] (AI)[104]
$P^{A}_{1,80}$: (AI)[104] [includes] (reasoning)[86]
$P^{A}_{1,86}$: (Machine learning)[128] [belongs to] (AI)[104]

Derived propositions from the blocks:

Proposition$_{1,8}$($A_{1,70}$; $P^{A}_{1,80}$; $P^{A}_{1,86}$) : {System intellect is provided by AI that includes reasoning and machine learning}.

Cluster 15:

	E54	E58	E50	E43	E56	E48	E46	E44	E39
E119	0	581	0	0	0	0	0	0	0
E52	0	0	501	0	0	0	0	0	0
E51	0	0	500	0	0	0	0	0	0
E42	0	0	0	420	0	0	0	0	0
E62	0	0	0	430	0	0	0	0	0
E55	540	552	0	0	550	0	0	0	0
E49	0	0	0	0	0	480	0	0	0
E47	0	0	0	0	0	0	460	0	0
E44	0	0	0	0	0	0	0	440	0
E40	0	0	0	0	0	0	0	0	390

Axioms and postulates mentioned in this cluster are:

$P^{D}_{1,3}$: (Toolbox)[58] [should learn from its] (application)[119]
$A_{1,31}$: (Complex outcomes)[50] [cause] (difficult interpretation)[52]
$P^{A}_{1,32}$: (Complex outcomes)[50] [are] (outcomes)[51]
$P^{A}_{1,28}$: (Heterogeneous user interface)[42] [is a] (user interface)[43]
$P^{A}_{1,40}$: (User interface)[43] [in an] (interface)[62]
$P^{A}_{1,34}$: (Software tools)[54] [belong to] (computational tools)[55]
$P^{A}_{1,37}$: (Toolboxes)[58] [belong to] (computational tools)[55]
$P^{A}_{1,35}$: (Software packages)[56] [belong to] (computational tools)[55]
$P^{A}_{1,31}$: (Different design tasks)[48] [are] (design tasks)[49]
$P^{A}_{1,30}$: (Instruction information)[46] [is an] (information)[47]
$P^{A}_{1,29}$: (Complex programming)[44] [is a] (programming)[45]
$P^{A}_{1,27}$: (Good training)[39] [is a] (training)[40]

Derived propositions from the blocks:

Proposition$_{1,8}$($P^{D}_{1,3}$; $P^{A}_{1,37}$) :{ Toolbox is a computational tool, and it should learn from its applications}.

Cluster 16:

	E106	E54	...	E21	E99
E103	1032	0	...	0	992

Axioms and postulates mentioned in this cluster are:

$A_{1,72}$: (System intellect)[103] [is provided by] (context information processing)[106]
$A_{1,69}$: (Smart semantics)[99] [uses] (system intellect)[103]

Derived propositions from the blocks:

Proposition$_{1,9}$($A_{1,72}$; $A_{1,69}$) : {Smart semantics uses system intellect that is provided by context information processing}.

Cluster 17:

	E110	E102	E24	...	E21	E99	E101
E111	1100	1020	0	...	0	994	1012

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,68}$: (PLC data meaning)[110] [is a] (data meaning)[111]
$P^{A}_{1,67}$: (MoL data meaning)[102] [is a] (data meaning)[111]
$A_{1,76}$: (Smart semantics)[99] [extracts] (data meaning)[111]
$A_{1,77}$: (Procedural reasoning)[101] [extracts] (data meaning)[111]

Derived propositions from the blocks:

Proposition$_{1,10}$($P^{A}_{1,67}$; $A_{1,76}$; $A_{1,77}$) : {Smart semantics and procedural reasoning extract the meaning of MoL data and PLC data }.

Cluster 18:

	E2	E21	E99	E101	E3	E98	E5	E65
E23	0	211	0	0	0	0	0	0
E103	0	0	992	0	0	0	0	0
E100	0	0	990	0	0	0	0	0
E111	0	0	994	1012	0	0	0	0
E110	0	0	993	1011	0	0	0	0
E102	0	0	991	1010	0	0	0	0
E24	0	0	0	0	32	0	0	0
E22	20	210	0	0	31	0	0	0
E92	0	0	0	0	0	924	0	0
E53	0	0	0	0	0	530	0	0
E86	0	0	0	860	0	0	850	0
E66	0	0	0	0	0	0	0	650

Axioms and postulates mentioned in this cluster are:

$A_{1,2}$: (Knowledge)[2] [is generated for] (product enhancement)[22]
$P^{A}_{1,16}$: (Data analytics method)[21] [is a] (method)[23]
$A_{1,12}$: (Data analytics methods)[21] [are used for] (product enhancement)[22]
$A_{1,69}$: (Smart semantics)[99] [uses] (system intellect)[103]
$P^{A}_{1,65}$: (Smart semantics)[99] [are] (semantics)[100]
$A_{1,76}$: (Smart semantics)[99] [extracts] (data meaning)[111]
$A_{1,66}$: (Smart semantics)[99] [extracts] (PLC data meaning)[110]
$A_{1,65}$: (Smart semantics)[99] [extracts] (MoL data meaning)[102]
$A_{1,77}$: (Procedural reasoning)[101] [extracts] (data meaning)[111]
$A_{1,68}$: (Procedural reasoning)[101] [extracts] (PLC data meaning)[110]
$A_{1,67}$: (Procedural reasoning)[101] [extracts] (MoL data meaning)[102]
$P^{A}_{1,66}$: (Procedural reasoning)[101] [is a] (reasoning)[86]
$P^{A}_{1,17}$: (Smart product)[24] [is a] (product)[3]
$P^{A}_{1,15}$: (Product enhancement)[22] [concerns] (products)[3]
$A_{1,62}$: (Sophisticated data management)[92] [facilitates] (human interpretation)[98]
$P^{A}_{1,63}$: (Human interpretation)[98] [is an] (interpretation)[53]
$P^{A}_{1,55}$: (Case-based reasoning)[85] [is a] (reasoning)[86]
$A_{1,41}$: (A complete software tool)[65] [has] (high performances)[66]

Derived propositions from the blocks:

Proposition$_{1,12}$($A_{1,2}$; $A_{1,12}$; $P^{A}_{1,15}$) : { data analytics methods generate Knowledge for product enhancement }.

Proposition$_{1,13}$($P^{A}_{1,68}$; $A_{1,76}$; $A_{1,77}$) : {Smart semantics and procedural reasoning extract meaning of PLC and MoL data }.

Cluster 19:

	E75	E73	...	E3	E98
E92	750	0	...	0	924

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,60}$: (Sophisticated data management)^{92} [belong to] (data management)^{75}
$A_{1,62}$: (Sophisticated data management)^{92} [facilitates] (human interpretation)^{98}

Derived propositions from the blocks:

Proposition$_{1,14}$($P^{A}_{1,60}$; $A_{1,62}$) : {Sophisticated data management facilitating human interpretation}.

Cluster 20:

	E76	E74	E72	E69
E77	760	0	0	0
E75	0	740	0	0
E73	0	0	720	0
E70	0	0	0	690

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,49}$: (Blended datasets)^{76} [are] (datasets)^{77}
$P^{A}_{1,48}$: (Multi-channel data management)^{74} [belongs to] (data management)^{75}
$P^{A}_{1,47}$: (Multifold data visualization)^{72} [is a] (data visualization)^{73}
$P^{A}_{1,45}$: (Step by step assistance)^{69} [is a] (guided assistance)^{70}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 21:

	E85	E65	E122	E84	E83	E82	E81	E76	...	E69	E68	E18	...	E64	E1
E54	0	541	0	0	0	544	543	0	...	0	542	0	...	0	-13
E58	580	0	0	0	0	0	0	0	...	0	0	0	...	0	-15

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,42}$: (Complete software tool)^{65} [is a] (software tool)^{54}
$P^{A}_{1,52}$: (Customized software tool)^{82} [is a] (software tool)^{54}
$P^{A}_{1,51}$: (Permanently accessible software tool)^{81} [is a] (software tool)^{54}
$P^{A}_{1,44}$: (Affordable software tool)^{68} [is a] (software tool)^{54}
$A_{1,32}$: (Data analytics tools)^{1} [include] (software tools)^{54}
$A_{1,34}$: (Data analytics tools)^{1} [include] (toolboxes)^{58}
$P^{D}_{1,4}$: (Toolbox)^{58} [should include] (case-based reasoning)^{85}

Derived propositions from the blocks:

$Proposition_{1,15}(A_{1,34}; P^{D}_{1,4})$: {Data analytics tools include software tools and toolboxes which should include case-based reasoning}.

$Proposition_{1,16}(P^{A}_{1,42}; P^{A}_{1,52}; P^{A}_{1,51}; P^{A}_{1,44}; A_{1,32})$: {Complete, customized, permanently accessible and affordable software tools are included in data analytics tools}.

Cluster 22:

	E53	E86	...	E58	E50
E52	520	0	...	0	501

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,33}$: (Difficult interpretation)[52] [is an] (interpretation)[53]
$A_{1,31}$: (Complex outcomes)[50] [cause] (difficult interpretation)[52]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 23:

	E5	E65
E86	850	0
E66	0	650
E77	0	0
...	...	...
E106	0	0
E54	0	541
E58	580	0

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,55}$: (Case-based reasoning)[85] [is a] (reasoning)[86]
$P^{D}_{1,4}$: (Toolbox)[58] [should include] (case-based reasoning)[85]
$A_{1,41}$: (A complete software tool)[65] [has] (high performances)[66]
$P^{A}_{1,42}$: (Complete software tool)[65] [is a] (software tool)[54]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 24:

	E18	E118	E33
E19	180	0	0
E34	0	0	330

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,14}$: (Product developer)[18] [is a] (developer)[19]

$P^{A}_{1,23}$: (Learning curve)^{33} [is a] (curve)^{34}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 25:

	E8	E17	E1	E37	E64
E7	70	0	0	0	0
E14	81	140	0	0	0
E13	80	0	130	0	0
E127	0	0	0	0	641
E106	0	0	0	0	640

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,3}$: (MoL)^{7} [belongs to] (PLC)^{8}
$P^{A}_{1,5}$: (PLC)^{8} [includes] (BoL)^{14}
$P^{A}_{1,4}$: (PLC)^{8} [includes] (EoL)^{13}
$P^{A}_{1,10}$: (BoL data)^{17} [is collected during] (BoL)^{14}
$P^{A}_{1,9}$: (EoL data)^{16} [is collected during] (EoL)^{13}
$P^{A}_{1,85}$: (Natural language processing)^{127} [is a] (processing)^{64}
$P^{A}_{1,73}$: (Context information processing)^{106} [is] (processing)^{64}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 26:

	E1	E20	E4	E35	E59	E15	E87	E63	E12	E78	E61	60
E54	-13	0	0	0	0	0	0	0	0	0	0	0
E58	-15	0	0	0	0	0	0	0	0	0	0	0
E50	-12	0	0	0	0	0	0	0	0	0	0	0
E43	18	0	0	0	0	0	0	0	0	0	0	0
E56	-14	0	0	0	0	0	0	0	0	0	0	0
E48	-11	0	0	0	0	0	0	0	0	0	0	0
E46	-10	0	0	0	0	0	0	0	0	0	0	0
E45	19	0	0	0	0	0	0	0	0	0	0	0
E39	17	0	0	0	0	0	0	0	0	0	0	0
E36	15	0	0	0	0	0	0	0	0	0	0	0
E2	10	0	0	0	0	0	0	0	0	0	0	0
E21	13	200	0	0	0	0	0	0	0	0	0	0
E99	0	0	-45	0	0	0	0	0	0	0	0	0
E101	0	0	-46	0	0	0	0	0	0	0	0	0
E3	0	0	30	0	0	0	0	0	0	0	0	0
E98	0	0	-44	0	0	0	0	0	0	0	0	0
E5	0	0	-42	0	0	0	0	0	0	0	0	0
E65	0	0	41	0	0	0	0	0	0	0	0	0
E122	0	0	-48	0	0	0	0	0	0	0	0	0
E84	-19	0	-41	0	0	0	0	0	0	0	0	0
E83	-18	0	-40	0	0	0	0	0	0	0	0	0
E82	0	0	49	0	0	0	0	0	0	0	0	0
E81	0	0	48	0	0	0	0	0	0	0	0	0
E76	0	0	46	0	0	0	0	0	0	0	0	0
E74	0	0	45	0	0	0	0	0	0	0	0	0
E72	0	0	44	0	0	0	0	0	0	0	0	0
E69	0	0	43	0	0	0	0	0	0	0	0	0
E68	0	0	42	0	0	0	0	0	0	0	0	0
E18	0	0	40	0	0	0	0	0	0	0	0	0
E118	0	0	-47	0	0	0	0	0	0	0	0	0
E33	14	0	0	331	0	0	0	0	0	0	0	0
E38	16	0	0	350	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	82	0	0	0	0	0	0
E17	0	0	0	0	0	151	0	0	0	0	0	0
E16	0	0	0	0	0	150	0	0	0	0	0	0
E37	0	0	0	0	0	0	370	0	0	0	0	0
E64	0	0	0	0	0	0	0	630	0	0	0	0
E1	10000	0	11	0	-16	0	0	-17	12	0	0	0
E20		0	0	0	0	0	0	0	120	0	0	0
E4			0	0	0	0	-43	0	0	47	0	0
E35				0	0	0	0	0	0	0	351	0
E59					0	0	0	0	0	0	0	590

Axioms and postulates mentioned in this cluster are:

$A_{1,32}$: (Data analytics tools)[1] [include] (software tools)[54]
$A_{1,30}$: (Data analytics tools)[1] [provide] (complex outcomes)[50]
$A_{1,38}$: (User interface)[43] of (data analytics tools)[1] [is complex]
$A_{1,33}$: (Data analytics tools)[1] [include] (software packages)[56]
$A_{1,29}$: (Data analytics tools)[1] [are not adaptable to] (different design tasks)[48]

$A_{1,28}$: (Instruction information)^{46} [is bad within] (data analytics tools)^{1}
$A_{1,39}$: (Programming)^{45} within (data analytics tools)^{1} [is complex]
$A_{1,25}$: (Good training)^{39} [is absent for] (data analytics tools)^{1}
$A_{1,1}$: (Data analytics tools)^{1} [generate] (knowledge)^{2}
$A_{1,10}$: (Data analytics tools)^{1} [exploit] (data analytics methods)^{21}
$P^{A}_{1,54}$: (Smart data analytics tool)^{84} [belongs to] (data analytics tools)^{1}
$P^{A}_{1,53}$: (Intuitive data analytics tool)^{83} [belongs to] (data analytics tools)^{1}
$A_{1,22}$: (Learning curve)^{33} of (data analytics tools)^{1} [is bad]
$A_{1,23}$: (Customer support)^{36} of (data analytics tools)^{1} [is bad]
$A_{1,24}$: (Learning time)^{38} of (data analytics tools)^{1} [is slow]
$A_{1,36}$: (Data analytics tools)^{1} [are not intuitive]
$P^{D}_{1,5}$: (Data analytics tool)^{1} [should be intuitive]
$P^{D}_{1,6}$: (Data analytics tool)^{1} [should be smart]
$A_{1,34}$: (Data analytics tools)^{1} [include] (toolboxes)^{58}
$A_{1,11}$: (Data analytics packages)^{20} [exploit] (data analytics methods)^{21}
$A_{1,63}$: (Designers)^{4} [need] (smart semantics)^{99}
$A_{1,64}$: (Designers)^{4} [need] (procedural reasoning)^{101}
$A_{1,3}$: (Designers)^{4} [enhance] (products)^{3}
$P^{A}_{1,67}$: (MoL data meaning)^{102} [is a] (data meaning)^{111}
$A_{1,54}$: (Designers)^{4} [need] (case-based reasoning)^{85}
$P^{A}_{1,78}$: (Designers)^{4} [create] (new products)^{c,78}
$A_{1,53}$: (Designers)^{4} [need] (smart data analytics tool)^{84}
$A_{1,50}$: (Designers)^{4} [need a] (permanently accessible software tool)^{81}
$A_{1,51}$: (Designers)^{4} [need a] (customized software tool)^{82}
$A_{1,52}$: (Designer)^{4} [need an] (intuitive data analytics tool)^{83}
$A_{1,46}$: (Designers)^{4} [need] (blended datasets)^{76}
$A_{1,45}$: (Designers)^{4} [need] (multi-channel data management)^{74}
$A_{1,44}$: (Designers)^{4} [need] (multifold data visualization)^{72}
$A_{1,43}$: (Designers)^{4} [need] (step by step assistance)^{69}
$A_{1,42}$: (Designers)^{4} [need an] (affordable software tool)^{68}
$A_{1,4}$: (Designers)^{4} [are] (product developers)^{18}
$P^{D}_{1,2}$: (Designers)^{4} [want to be advised in their] (choices)^{118}
$A_{1,78}$: (Designers)^{4} [use] (data analytics tools)^{1}
$P^{D}_{1,1}$: (Designers)^{4} [want to have a] (complete software tool)^{65}
$P^{A}_{1,24}$: (Learning curve)^{33} [is proportional with] the (time)^{35}
$P^{A}_{1,26}$: (Learning time)^{38} [is a] (time)^{35}
$A_{1,35}$: (Data analytics tools)^{1} [contain] (unsolved bugs)^{59}
$P^{A}_{1,8}$: (PLC data)^{15} [is collected during] (PLC)^{8}
$P^{A}_{1,7}$: (PLC data)^{15} [include] (BoL data)^{17}
$P^{A}_{1,6}$: (PLC data)^{15} [include] (EoL data)^{16}
$P^{A}_{1,56}$: (Semantic support)^{87} [is a] (support)^{37}
$P^{D}_{1,7}$: (Designers)^{4} [want to have] (semantic support)^{87}
$P^{A}_{1,41}$: (Data processing)^{63} [is a] (processing)^{64}
$A_{1,40}$: (Data processing)^{63} within (data analytics tools)^{1} [is complex]
$A_{1,8}$: (Data analytics tools)^{1} [extract] (product-associated data)^{12}
$A_{1,9}$: (Data analytics packages)^{20} [extract] (product-associated data)^{12}
$A_{1,47}$: (Designers)^{4} [need] (combined data)^{78}
$A_{1,37}$: (Time)^{35} (is consumed in collecting) (relevant data)^{61}
$P^{A}_{1,38}$: (Unsolved bug)^{59} [is a] (bug)^{60}

Derived propositions from the blocks:

$Proposition_{1,17}(A_{1,36}; A_{1,52})$: {Data analytics tools are not intuitive but designers need intuitive data analytics tools}.

$Proposition_{1,18}(A_{1,28}; A_{1,22}; A_{1,23})$:{Instruction information, customer support and learning curve are bad within data analytics tools}.

$Proposition_{1,19}(A_{1,32}; A_{1,33}; A_{1,34})$: Data analytics tools include software tools and packages as well as toolboxes.

$Proposition_{1,20}(A_{1,38}; A_{1,39}; A_{1,24}; A_{1,35}; A_{1,40}; A_{1,30})$: {Data processing, programming and user interface are complex within data analytics tools and have a slow learning time, contain unsolved bugs and an absent good training and provide complex outcomes}.

$Proposition_{1,21}(A_{1,28}; A_{1,78})$: {Designers use data analytics tools which are not adaptable to different design tasks}.

$Proposition_{1,22}(P^{D}_{1,5}; P^{D}_{1,6}; A_{1,53})$: {Designers need smart data analytics tools. They should be smart and intuitive}.

$Proposition_{1,23}(A_{1,1}; A_{1,8}; A_{1,9}; A_{1,10}; A_{1,11})$: {Data analytics tools and packages exploit data analytics methods which extract product-associated data and generate knowledge}.

$Proposition_{1,24}(A_{1,78}; A_{1,42}; A_{1,50}; A_{1,51}; P^{D}_{1,1})$: {Designers use data analytics tools and need a complete, affordable, permanently accessible and customized software tool}.

$Proposition_{1,25}(A_{1,4}; A_{1,3}; P^{A}_{1,78})$: {Designers are product developers, they enhance products and create new ones}.

$Proposition_{1,26}(A_{1,63}; P^{D}_{1,7})$: {Designers need smart semantics and want to have semantic support}.

$Proposition_{1,27}(A_{1,37}; A_{1,47}; A_{1,46})$: {Designers need combined data and blended datasets, and time is consumed in collecting relevant data}.

$Proposition_{1,28}(A_{1,64}; A_{1,54})$: {Designers need procedural and case-based reasoning}.

$Proposition_{1,29}(A_{1,45}; A_{1,44})$: {Designers need multi-channel data management and multifold data visualization}.

$Proposition_{1,30}(A_{1,43}; P^{D}_{1,2})$: {Designers need step by step assistance and want to be advised in their choices}.

Cluster 27:

	E37	E64	E1
E36	360	0	15

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,25}$: (Customer support)^{36} [is a] (support)^{37}
$A_{1,23}$: (Customer support)^{36} of (data analytics tools)^{1} [is bad]

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 28:

	E122	E84	...	E20	E4
E3	33	0	...	0	30

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,79}$: (New product)^{122} [is a] (product)^{3}
$A_{1,3}$: (Designers)^{4} [enhance] (products)^{3}

Derived propositions from the blocks:

No propositions derived from this cluster.

Cluster 29:

	E9
E15	91
E87	97
E63	93
E12	90
E78	94
E61	0
E60	92
E80	96
E79	95

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,11}$: (PLC data)^{15} [are] (data)^{9}
$A_{1,55}$: (Semantic support)^{87} [is provided to] (data)^{9}
$P^{A}_{1,87}$: (Data processing)^{63} [processes] (data)^{9}
$P^{A}_{1,88}$: (Product-accociated data)^{12} [is a] (data)^{9}
$P^{A}_{1,50}$: (Combined data)^{78} [group] (data)^{9}
$P^{A}_{1,39}$: (Relevant data)^{61} [are] (data)^{9}
$A_{1,49}$: (Quantitative data)^{80} [is included in] (data)^{9}
$A_{1,48}$: (Qualitative data)^{79} [is included in] (data)^{9}

Derived propositions from the blocks:

Proposition$_{1,31}$($A_{1,55}$; $A_{1,49}$; $A_{1,48}$) :{Semantic support is provided to qualitative and quantitative data, to product-associated data}.

Proposition$_{1,33}$($P^{A}_{1,50}$; $A_{1,49}$; $A_{1,48}$) : {Combined data group qualitative and quantitative data}.

Cluster 30:

	E123	E90	E121	E120	E88	E89
E9	98	0	0	0	0	0
E91	0	900	0	0	0	0
E123	0	0	0	1201	0	0
E90		0	0	901	0	0
E121			0	1200	0	0
E120				0	881	0
E88					880000	880

Axioms and postulates mentioned in this cluster are:

$P^{A}_{1,59}$: (Incomplete data)^{123} [belong to] (data)^{9}
$P^{A}_{1,58}$: (Analytical need)^{90} [is a] (need)^{91}
$P^{D}_{1,9}$: (Data analytics solutions)^{120} [should be in harmony with] (analytical needs)^{90}
$P^{D}_{1,10}$: (Data analytics solutions)^{120} [should cope with] (incomplete data)^{123}
$P^{A}_{1,77}$: (Data analytics solutions)^{120} [are] (solutions)^{121}
$P^{D}_{1,8}$: (Data analytics solutions)^{120} [should be in harmony with] (data collection practices)^{88}
$A_{1,56}$: (Data collection practices)^{88} [are multiple]
$A_{1,57}$: (Data collection practices)^{88} [are heterogeneous]
$P^{A}_{1,57}$: (Data collection practice)^{88} [is a] (practice)^{89}

Derived propositions from the blocks:

Proposition$_{1,34}$($P^{D}_{1,9}$; $P^{D}_{1,10}$; $P^{A}_{1,58}$; $P^{A}_{1,59}$; $P^{A}_{1,77}$; $P^{D}_{1,8}$; $A_{1,56}$; $A_{1,57}$; $P^{A}_{1,57}$) : {Data analytics solutions should cope with incomplete data, and be in harmony with analytical needs and data collection practices that are multiple and heterogeneous}.

Appendix 3

Global set of entities of the combined five theories

Legend:

$E_{x,y}$: refers to the derived entities, where “x” is the number of the theory and “y” is the order of appearance of the entity in the textual formulation of the theory x.
$E_{c,z}$: refers to the common entities, where “z” is the order of detection of the common theory.

List of abbreviation:

Abbreviation	Designation	Abbreviation	Designation
ANN	Artificial neural network	AI	Artificial intelligence
BoL	Beginning-of-life	CI	Computational intelligence
CNN	Convolutional neural network	CPS	Cyber-physical-system
DL	Deep learning	DNN	Deep neural network
EA	Evolutionary algorithm	EoL	End-of-life
FL	Fuzzy logic	HCI	Human-computer interaction
HHI	Human-human interaction	HMI	Human-machine interaction
HSI	Human-system interaction	HTI	Human-tool interaction
ML	Machine learning	MoL	Middle-of-life
NN	Neural network	PLC	Product life cycle
SHI	System-human interaction	SSI	System-system interaction

Global set of theories:

Entity code	Denomination	Entity code	Denomination	Entity code	Denomination
$E_{c,1}$	Data analytics tool	$E_{c,2}$	Knowledge	$E_{c,3}$	Product
$E_{c,62}$	Designer	$E_{1,5}$	Use data	$E_{1,6}$	MoL data
$E_{1,7}$	MoL	$E_{1,8}$	PLC	$E_{c,4}$	Data

$E_{1,10}$	Work data	$E_{1,11}$	Failure data	$E_{1,12}$	Product-associated data
$E_{1,13}$	EoL	$E_{1,14}$	BoL	$E_{1,15}$	PLC data
$E_{1,16}$	EoL data	$E_{1,17}$	BoL data	$E_{1,18}$	Product developer
$E_{1,19}$	Developer	$E_{1,20}$	Data analytics package	$E_{c,5}$	Data analytics method
$E_{1,22}$	Product enhancement	$E_{c,6}$	Method	$E_{1,24}$	Smart product
$E_{1,25}$	Self-learning capability	$E_{1,26}$	Self-adaptive capability	$E_{1,27}$	Self-management capability
$E_{1,28}$	Capability	$E_{1,29}$	Operational state	$E_{c,7}$	State
$E_{1,31}$	Use circumstance	$E_{1,32}$	Circumstance	$E_{1,33}$	Learning curve
$E_{1,34}$	Curve	$E_{c,8}$	Time	$E_{1,36}$	Customer support
$E_{1,37}$	Support	$E_{1,38}$	Learning time	$E_{1,39}$	Good training
$E_{1,40}$	Training	$E_{1,41}$	Lack of ease of use	$E_{1,42}$	Heterogeous user interface
$E_{1,43}$	User interface	$E_{1,44}$	Complex programming	$E_{c,9}$	Programming
$E_{1,46}$	Instruction information	$E_{c,10}$	Information	$E_{1,48}$	Different design task
$E_{c,64}$	Design task	$E_{1,50}$	Complex outcome	$E_{c,11}$	Outcome
$E_{1,52}$	Difficult interpretation	$E_{c,72}$	Interpretation	$E_{1,54}$	Software tool
$E_{c,12}$	Computational tool	$E_{1,56}$	Software package	$E_{c,13}$	Tool
$E_{1,58}$	Toolbox	$E_{1,59}$	Unsolved bug	$E_{1,60}$	Bug
$E_{1,61}$	Relevant data	$E_{1,62}$	Interface	$E_{c,14}$	Data processing
$E_{c,15}$	Processing	$E_{1,65}$	Complete software tool	$E_{c,16}$	High performance
$E_{c,17}$	Performance	$E_{1,68}$	Affordable software tool	$E_{1,69}$	Step by step assistance
$E_{1,70}$	Guided assistance	$E_{1,71}$	Assistance	$E_{1,72}$	Multifold data visualization
$E_{c,18}$	Data visualization	$E_{1,74}$	Multi-channel data management	$E_{1,75}$	Data management
$E_{1,76}$	Blended datasets	$E_{c,19}$	Dataset	$E_{1,78}$	Combined data
$E_{c,20}$	Qualitative data	$E_{1,80}$	Quantitative data	$E_{1,81}$	Permanently accessible software tool
$E_{1,82}$	Customized software tool	$E_{1,83}$	Intuitive data analytics tool	$E_{1,84}$	Smart data analytics tool
$E_{1,85}$	Case-based reasoning	$E_{1,86}$	Reasoning	$E_{c,66}$	Semantic support
$E_{c,21}$	Data collection practice	$E_{1,89}$	Practice	$E_{1,90}$	Analytical need
$E_{c,22}$	Need	$E_{1,92}$	Sophisticated data management	$E_{1,93}$	Data stream
$E_{1,94}$	Data fusion	$E_{1,95}$	Computational performance	$E_{1,96}$	Fusion
$E_{1,97}$	Usability	$E_{1,98}$	Human interpretation	$E_{1,99}$	Smart semantics

$E_{1,100}$	Semantics	$E_{1,101}$	Procedural reasoning	$E_{1,102}$	MoL data meaning
$E_{1,103}$	System intellect	$E_{1,104}$	AI	$E_{1,105}$	System learning mechanism
$E_{1,106}$	Context information processing	$E_{1,107}$	Situation awareness	$E_{1,108}$	Strategy development
$E_{1,109}$	System adaptation capability	$E_{1,110}$	PLC data meaning	$E_{c,23}$	Data meaning
$E_{c,112}$	Meaning	$E_{c,60}$	Intelligence	$E_{1,114}$	Learning mechanism
$E_{c,24}$	Mechanism	$E_{c,68}$	Awareness	$E_{c,25}$	Development
$E_{1,118}$	Choice	$E_{c,26}$	Application	$E_{1,120}$	Data analytics solution
$E_{c,27}$	Solution	$E_{c,78}$	New product	$E_{1,123}$	Incomplete data
$E_{1,124}$	Knowledge representation	$E_{1,125}$	Representation	$E_{c,28}$	Learning
$E_{1,127}$`	Natural language processing	$E_{c,59}$	Machine learning	$E_{c,80}$	Task
$E_{c,29}$	Big data	$E_{c,30}$	Volume	$E_{2,3}$	Huge data amount
$E_{c,31}$	Velocity	$E_{2,5}$	Data creation speed	$E_{2,6}$	Data stream speed
$E_{2,7}$	Data aggregation speed	$E_{2,8}$	Data movement speed	$E_{c,32}$	Variety
$E_{c,33}$	Data type	$E_{c,34}$	Structured data	$E_{c,35}$	Semi-structured data
$E_{c,36}$	Unstructured data	$E_{c,37}$	Veracity	$E_{2,15}$	Data messiness
$E_{2,16}$	Data trustworthiness	$E_{c,38}$	Value	$E_{c,39}$	Big data analytics
$E_{2,20}$	Hidden pattern	$E_{2,21}$	Relationship	$E_{2,22}$	Application context
$E_{2,23}$	Advanced big data analytics	$E_{2,24}$	Intelligent computerized solution	$E_{2,25}$	Sentiment analysis
$E_{2,27}$	Service	$E_{2,28}$	User's opinion	$E_{2,29}$	Evaluation
$E_{2,30}$	Affective state	$E_{c,40}$	Organization	$E_{2,32}$	Cloud computing service
$E_{2,33}$	Big data analysis requirement	$E_{c,58}$	Diverse data sources	$E_{2,36}$	Online data processing
$E_{2,37}$	Signal input	$E_{2,38}$	Real-life application	$E_{2,39}$	High speed storage
$E_{2,40}$	High speed data processing	$E_{2,41}$	Big data analytics method	$E_{2,42}$	Interpretable knowledge
$E_{2,43}$	Big data analytics technique	$E_{2,44}$	Online adaptation	$E_{2,45}$	Contextual element
$E_{2,46}$	User-specific element	$E_{2,47}$	Design	$E_{2,48}$	Decision-making mechanism
$E_{c,41}$	Computational technique	$E_{2,50}$	Machine learning technique	$E_{2,52}$	Patterns modeling
$E_{2,53}$	Correlations modeling	$E_{c,65}$	Prediction	$E_{2,55}$	Unseen data
$E_{c,79}$	Event	$E_{2,57}$	Supervised learning	$E_{2,58}$	Labelled data
$E_{2,59}$	Unsupervised learning	$E_{2,60}$	Reinforcement learning	$E_{2,61}$	Goal oriented learning

$E_{2,62}$	Dynamic situation	$E_{c,42}$	Regression technique	$E_{c,43}$	Clustering approach
$E_{2,65}$	Density estimation method	$E_{2,66}$	Dimensionality reduction approach	$E_{c,44}$	Algorithm
$E_{2,69}$	Human information processing mechanism	$E_{2,70}$	Human information reasoning mechanism	$E_{2,71}$	Computational intelligence technique
$E_{2,72}$	Real-world data driven problem	$E_{2,73}$	Mathematical modeling	$E_{2,74}$	Traditional modeling
$E_{2,75}$	Highly complex process	$E_{2,76}$	Highly uncertain process	$E_{2,77}$	Highly stochastic process
$E_{2,78}$	FL	$E_{2,79}$	EA	$E_{2,80}$	ANN
$E_{2,81}$	CI approach	$E_{2,82}$	Methodology	$E_{2,83}$	Imprecise data
$E_{2,84}$	Uncertain data	$E_{2,86}$	Adaptive control	$E_{2,87}$	Linguistic qualifier
$E_{2,88}$	Fuzzy set	$E_{2,89}$	Uncertain real-world	$E_{2,90}$	Uncertain user defined concept
$E_{2,91}$	Human interpretable fuzzy rule	$E_{2,92}$	Inference	$E_{c,61}$	Decision-making
$E_{2,94}$	Genetic algorithm	$E_{2,95}$	Genetic processing	$E_{2,96}$	Swarm intelligence optimization algorithm
$E_{2,97}$	Complex real-world system	$E_{2,98}$	Complex real-world processes	$E_{2,99}$	Feature extraction
$E_{2,100}$	Experiential data	$E_{2,101}$	CI techniques combination	$E_{c,45}$	Insight
$E_{2,103}$	Integrated solution	$E_{2,104}$	Offline data processing	$E_{2,105}$	Effective multipurpose intelligent data analysis
$E_{2,106}$	Effective decision-making	$E_{2,108}$	Important feature identification	$E_{2,109}$	Important feature
$E_{c,46}$	Output	$E_{2,111}$	Spatial co-relations identification	$E_{2,112}$	Spatial co-relations
$E_{2,113}$	Given time	$E_{2,114}$	Input variable	$E_{2,115}$	Temporal co-relations identification
$E_{2,116}$	Temporal co-relations	$E_{2,117}$	Input parameter	$E_{2,118}$	Overtime
$E_{2,119}$	DL approach	$E_{2,120}$	Large-scale data	$E_{2,121}$	Spatial correlation
$E_{2,122}$	Temporal correlation	$E_{2,123}$	Multiple hidden layers	$E_{2,124}$	Feature learning method
$E_{2,125}$	Supervised approach	$E_{2,126}$	DNN	$E_{2,127}$	CNN
$E_{2,128}$	Recurrent NN	$E_{2,129}$	DL technique	$E_{2,130}$	Pattern recognition
$E_{2,131}$	Computationally expensive	$E_{2,132}$	Training time	$E_{2,133}$	Natural language fuzzy rule
$E_{2,134}$	Hidden relation	$E_{c,47}$	Pattern	$E_{2,136}$	User-friendly system
$E_{2,137}$	Excellent data analysis tool	$E_{2,138}$	High dimensionality	$E_{2,139}$	Sparseness

$E_{2,140}$	Data clustering	$E_{2,141}$	Feature selection	$E_{2,142}$	Various data types
$E_{2,143}$	Complex data source	$E_{2,144}$	Uncertain data source	$E_{2,145}$	Complex real-world driven problem
$E_{2,146}$	Variable signal input	$E_{2,147}$	Diverse data types	$E_{2,148}$	Data amount
$E_{c,67}$	Context	$E_{c,48}$	Data analytics	$E_{2,151}$	Computized solution
$E_{c,49}$	Analysis	$E_{2,154}$	Opinion	$E_{2,156}$	Computing service
$E_{2,157}$	Requirement	$E_{2,159}$	Input	$E_{2,161}$	Storage
$E_{2,162}$	Adaptation	$E_{2,163}$	Element	$E_{c,50}$	Technique
$E_{2,166}$	Approach	$E_{2,167}$	Computational approach	$E_{c,51}$	Modeling
$E_{2,169}$	Correlation	$E_{2,171}$	Situation	$E_{2,173}$	Estimation method
$E_{2,174}$	Processing mechanism	$E_{2,175}$	Reasoning mechanism	$E_{c,52}$	Data source
$E_{2,177}$	Data driven problem	$E_{c,77}$	Problem	$E_{c,53}$	Process
$E_{2,180}$	NN	$E_{2,181}$	Control	$E_{2,182}$	Qualifier
$E_{2,183}$	Real-world	$E_{2,184}$	User-defined concept	$E_{c,54}$	Concept
$E_{2,186}$	Fuzzy rule	$E_{c,76}$	Rule	$E_{2,189}$	Optimization algorithm
$E_{c,55}$	System	$E_{2,191}$	Real-world system	$E_{2,192}$	Real-world process
$E_{2,193}$	Intelligent data analysis	$E_{2,194}$	Data analysis	$E_{2,195}$	Feature identification
$E_{2,196}$	Feature	$E_{2,197}$	Co-relation identification	$E_{2,198}$	Relation-identification
$E_{c,56}$	Identification	$E_{2,200}$	Co-relation	$E_{c,71}$	Relation
$E_{2,202}$	Variable	$E_{2,204}$	Parameter	$E_{2,205}$	Hidden layer
$E_{2,206}$	Layer	$E_{2,207}$	Learning method	$E_{2,209}$	Dimensionality
$E_{c,57}$	Data analytics technique	$E_{2,213}$	Data analysis requirement	$E_{2,214}$	Reduction approach
$E_{2,215}$	Complex process	$E_{2,216}$	Uncertain process	$E_{2,217}$	Stochastic process
$E_{2,218}$	DL	$E_{3,3}$	Data organization	$E_{3,6}$	Digitalization
$E_{3,7}$	Data quality	$E_{3,8}$	Data variety	$E_{3,10}$	Data behavior
$E_{3,12}$	Qualitative technique	$E_{3,13}$	Quantitative technique	$E_{3,15}$	Customer preference
$E_{3,18}$	Descriptive analytics	$E_{3,19}$	Analytical technique	$E_{3,20}$	Predictive analytics
$E_{3,21}$	Prescriptive analytics	$E_{3,22}$	Historic data	$E_{3,23}$	Probability analysis
$E_{3,24}$	Trending	$E_{3,25}$	Data association development	$E_{3,28}$	Happening
$E_{c,63}$	Action	$E_{3,30}$	Happening time frame	$E_{3,31}$	Experience
$E_{3,32}$	Specific domain	$E_{3,33}$	Specific process	$E_{3,35}$	Analytics project
$E_{3,36}$	Problem definition	$E_{3,37}$	Gathering required data	$E_{3,38}$	Data pre-processing
$E_{3,39}$	Performing analytics over data	$E_{3,43}$	Data attribute	$E_{3,44}$	Data format
$E_{3,46}$	Algorithmic concept	$E_{3,48}$	Classification	$E_{3,50}$	Model-based recommendation

$E_{3,51}$	Data analytics output	$E_{3,52}$	Behavioral analytics	$E_{3,53}$	Social media analytics
$E_{3,54}$	Time series analysis	$E_{3,55}$	Video	$E_{3,56}$	Audio
$E_{3,57}$	Digital image	$E_{3,58}$	Sensor	$E_{3,59}$	Log file
$E_{3,60}$	Transactional application	$E_{3,61}$	Web	$E_{3,62}$	Social media
$E_{3,63}$	Real time	$E_{3,64}$	Large scale	$E_{3,69}$	Dataset size
$E_{3,71}$	Data complexity	$E_{3,72}$	Data generation speed	$E_{3,73}$	Data delivery speed
$E_{3,74}$	Data availability	$E_{3,76}$	Relational database management system	$E_{3,78}$	Structure
$E_{3,79}$	Scientific data	$E_{3,80}$	Bibliographic data	$E_{3,81}$	Graph data structure
$E_{3,83}$	Document	$E_{3,84}$	Multi-structured data	$E_{3,85}$	Data mix
$E_{3,86}$	Operating system level log	$E_{3,88}$	Monitoring	$E_{3,89}$	Challenge
$E_{3,90}$	Dimension	$E_{3,91}$	Data challenge	$E_{3,92}$	Process challenge
$E_{3,93}$	Management challenge	$E_{3,95}$	Uncertainty	$E_{3,96}$	Imprecision
$E_{3,98}$	Statement	$E_{3,99}$	Untruth	$E_{3,100}$	Data discovery
$E_{3,101}$	Data relevance	$E_{3,102}$	Data comprehensiveness	$E_{3,104}$	Data alignment
$E_{3,105}$	Data transformation	$E_{3,106}$	Data modeling	$E_{3,107}$	Result visualization
$E_{3,108}$	Result sharing	$E_{3,109}$	Data privacy	$E_{3,110}$	Data security
$E_{3,111}$	Data governance	$E_{3,112}$	Data processing system	$E_{3,113}$	Big data technology
$E_{3,114}$	Secure way	$E_{3,115}$	Data mining	$E_{3,116}$	Subsequent analysis
$E_{3,117}$	Large dataset	$E_{3,119}$	Traditional big data analytics technique	$E_{3,120}$	Raw data
$E_{3,121}$	Actionable insight	$E_{c,69}$	Behavior	$E_{3,124}$	Preference
$E_{3,128}$	New insight	$E_{3,129}$	Classified data association development	$E_{3,131}$	Categorized data association development
$E_{3,132}$	Future outcome	$E_{3,133}$	Time frame	$E_{3,135}$	Best possible outcome
$E_{3,136}$	Possible outcome	$E_{3,137}$	Domain	$E_{3,139}$	Project
$E_{3,143}$	Attribute	$E_{3,144}$	Fixed data format	$E_{3,146}$	Recommendation
$E_{3,147}$	Visualization	$E_{3,148}$	Advanced data analytics technique	$E_{3,149}$	Analytics
$E_{3,151}$	Scale	$E_{3,152}$	Speed	$E_{3,153}$	Definite pattern
$E_{3,154}$	Management system	$E_{3,156}$	Pre-defined structure	$E_{3,157}$	Different structure
$E_{3,158}$	Standard structure	$E_{3,159}$	Log	$E_{c,70}$	Huge information
$E_{3,161}$	Multiple data types	$E_{3,162}$	Multiple data sources	$E_{3,163}$	Multiple data formats
$E_{3,164}$	Missing value	$E_{3,165}$	Missing statement	$E_{3,166}$	Huge data challenge
$E_{3,167}$	High data quality	$E_{3,168}$	Huge dataset	$E_{c,75}$	Significant information
$E_{3,170}$	Traditional data processing system	$E_{3,173}$	Simplified structure	$E_{c,74}$	Technology

$E_{4,1}$	Design problem	$E_{4,3}$	Functional requirement	$E_{4,4}$	Constraint
$E_{4,8}$	Course of action	$E_{4,10}$	Impact	$E_{4,12}$	Creativity
$E_{4,13}$	Novelty	$E_{4,17}$	Operation	$E_{4,19}$	Design process
$E_{4,20}$	Decision quality	$E_{4,21}$	Level of proactivity	$E_{4,22}$	6R
$E_{4,23}$	Observation	$E_{4,25}$	Diagnostic	$E_{4,27}$	Prescription
$E_{4,29}$	Key influencing event	$E_{4,30}$	Appropriate inquiry	$E_{4,31}$	Mission
$E_{4,32}$	Environment	$E_{4,33}$	Asset	$E_{4,34}$	Task
$E_{4,35}$	Threat	$E_{4,36}$	Up-to-date data source	$E_{c,73}$	Objective
$E_{4,38}$	Workload	$E_{c,81}$	Role	$E_{4,42}$	Proactive decision-making
$E_{4,43}$	Context changes	$E_{4,44}$	Unmanned system	$E_{4,45}$	Smart reasoning technologies
$E_{4,47}$	Reasoning technology	$E_{4,48}$	Human	$E_{4,49}$	Autonomous gent
$E_{4,50}$	Resource	$E_{4,51}$	Decision	$E_{4,53}$	Product information
$E_{4,54}$	Product knowledge	$E_{4,55}$	Purposive novelty	$E_{4,56}$	Difficult design problem
$E_{4,57}$	Obvious solution	$E_{4,58}$	Creative solution	$E_{4,59}$	Knowledge-based system
$E_{4,60}$	High-level decision-making	$E_{4,61}$	Creative decision-making	$E_{4,64}$	Decision-making process
$E_{4,67}$	Robust decision-making	$E_{4,69}$	Inquiry	$E_{4,70}$	Context-driven decision-making
$E_{4,71}$	Dynamically integrated knowledge	$E_{4,73}$	Real-time data	$E_{4,75}$	Agent
$E_{4,76}$	Decision-support systems	$E_{4,77}$	New context	$E_{4,78}$	Actionable decision
$E_{5,1}$	HSI	$E_{5,2}$	Level of interaction	$E_{5,3}$	Domain of interaction
$E_{5,4}$	Context of interaction	$E_{5,5}$	Modality of interaction	$E_{5,6}$	HHI
$E_{5,7}$	HTI	$E_{5,8}$	HMI	$E_{5,9}$	HCI
$E_{5,10}$	Human role	$E_{5,13}$	SSI	$E_{5,14}$	Activity
$E_{5,16}$	Proactivity	$E_{5,17}$	CPS	$E_{5,18}$	Interaction
$E_{5,20}$	Forming the intention	$E_{5,22}$	Evaluating the outcomes	$E_{5,23}$	Mental activity
$E_{5,24}$	Factor	$E_{5,26}$	Level	$E_{5,27}$	Skill
$E_{5,30}$	Component	$E_{5,32}$	Connectivity	$E_{5,37}$	Actor
$E_{5,38}$	Ontology	$E_{5,40}$	interrelationship	$E_{5,41}$	Physical level of interaction
$E_{5,42}$	Syntactic level of interaction	$E_{5,43}$	Semantic level of interaction	$E_{5,44}$	Pragmatic level of interaction
$E_{5,45}$	Apobetic level of interaction	$E_{5,46}$	Perceptive domain of interaction	$E_{5,47}$	Cognitive domain of interaction

$E_{5,48}$	Motor domain of interaction	$E_{5,49}$	Emotional domain of interaction	$E_{5,51}$	Selecting an action
$E_{5,52}$	Executing an action	$E_{5,53}$	Physical activity	$E_{5,54}$	Human factor
$E_{5,55}$	Machine factor	$E_{5,56}$	Interaction factor	$E_{5,57}$	Human behaviour
$E_{5,58}$	Smart system	$E_{5,59}$	Adaptive system	$E_{5,60}$	Decentralized system
$E_{5,61}$	Distributed system	$E_{5,62}$	Multi-scale system	$E_{5,63}$	Varied component
$E_{5,64}$	Internal relation	$E_{5,65}$	External relation	$E_{5,66}$	Physical connectivity
$E_{5,67}$	Syntactic interaction	$E_{5,69}$	Human mental process	$E_{5,70}$	Semantic technology
$E_{5,71}$	Computational actor	$E_{5,72}$	CPSs interaction	$E_{5,73}$	Traditional HCI
$E_{5,74}$	Traditional HMI	$E_{5,75}$	Internal interrelationship	$E_{5,76}$	External interrelationship
$E_{5,77}$	Diverse interactions	$E_{5,78}$	Intelligent system	$E_{5,79}$	Intelligence-based SHI

Appendix 4

Global set of axioms and postulates derived from the five component theories

Legend:

$A_{x,y}$: refers to the derived axioms from the theory. “x” is the number of the theory and “y” is the order of appearance of the axiom in the textual formulation of the theory x.
$P^{D}_{x,y}$: refers to the derived postulates from the textual theory. “x” is the number of the theory and “y” is the order of appearance of the derived postulate in the textual formulation of the theory x.
$P^{A}_{1,x}$: refers to the auxiliary postulates based on personal knowledge. “x” is the number of the theory and “y” is the order of appearance of the building of the auxiliary postulate.
$(..)^{x}$: represent the entity with the order of appearance “x,” contained in the theory referred to by its corresponding axiom or postulate.
$(..)^{c,x}$: represent common entity with the order of appearance x.
[..] : represent the relationship between two entities

Derived axioms from the five component theories:

$A_{1,1}$: (Data analytics tools)^{c,1} [generate] (knowledge)^{c,2}
$A_{1,2}$: (Knowledge)^{c,2} [is generated for] (product enhancement)^{22}
$A_{1,3}$: (Designers)^{c,62} [enhance] (products)^{c,3}
$A_{1,4}$: (Designers)^{c,62} [are] (product developers)^{18}
$A_{1,5}$: (Product enhancement)^{22} [is based on] (use data)^{5}
$A_{1,6}$: (Product enhancement)^{22} [is based on] (work data)^{10}
$A_{1,7}$: (Product enhancement)^{22} [is based on] (failure data)^{11}
$A_{1,8}$: (Data analytics tools)^{c,1} [extract] (product-associated data)^{12}
$A_{1,9}$: (Data analytics packages)^{20} [extract] (product-associated data)^{12}
$A_{1,10}$: (Data analytics tools)^{c,1} [exploit] (data analytics methods)^{c,5}
$A_{1,11}$: (Data analytics packages)^{20} [exploit] (data analytics methods)^{c,5}
$A_{1,12}$: (Data analytics methods)^{c,5} [are used for] (product enhancement)^{22}
$A_{1,13}$: (Smart products)^{24} [incorporate] (self-learning capabilities)^{25}
$A_{1,14}$: (Smart products)^{24} [incorporate] (self-adaption capabilities)^{26}
$A_{1,15}$: (Smart products)^{24} [incorporate] (self-management capabilities)^{27}
$A_{1,16}$: (Smart products)^{24} [collect their] (operational state)^{29}

$A_{1,17}$: (Smart products)[24] [communicate their] (operational state)[29]
$A_{1,18}$: (Smart products)[24] [reason with their] (operational state)[29]
$A_{1,19}$: (Smart products)[24] [collect their] (use circumstances)[31]
$A_{1,20}$: (Smart products)[24] [communicate their] (use circumstances)[31]
$A_{1,21}$: (Smart products)[24] [reason with their] (use circumstances)[31]
$A_{1,22}$: (Learning curve)[33] of (data analytics tools)[c,1] [is bad]
$A_{1,23}$: (Customer support)[36] of (data analytics tools)[c,1] [is bad]
$A_{1,24}$: (Learning time)[38] of (data analytics tools)[c,1] [is slow]
$A_{1,25}$: (Good training)[39] [is absent for] (data analytics tools)[c,1]
$A_{1,26}$: (Lack of ease of use)[41] [is caused by] (heterogeneous user interfaces)[42]
$A_{1,27}$: (Lack of ease of use)[41] [is caused by] (complex programming)[44]
$A_{1,29}$: (Data analytics tools)[c,1] [are not adaptable to] (different design tasks)[48]
$A_{1,30}$: (Data analytics tools)[c,1] [provide] (complex outcomes)[50]
$A_{1,31}$: (Complex outcomes)[50] [cause] (difficult interpretation)[52]
$A_{1,32}$: (Data analytics tools)[c,1] [include] (software tools)[54]
$A_{1,33}$: (Data analytics tools)[c,1] [include] (software packages)[56]
$A_{1,34}$: (Data analytics tools)[c,1] [include] (toolboxes)[58]
$A_{1,35}$: (Data analytics tools)[c,1] [contain] (unsolved bugs)[59]
$A_{1,36}$: (Data analytics tools)[c,1] [are not intuitive]
$A_{1,37}$: (Time)[c,8] [is consumed in collecting] (relevant data)[61]
$A_{1,38}$: (User interface)[43] of (data analytics tools)[c,1] [is complex]
$A_{1,39}$: (Programming)[c,9] within (data analytics tools)[c,1] [is complex]
$A_{1,40}$: (Data processing)[c,14] within (data analytics tools)[c,1] [is complex]
$A_{1,41}$: (A complete software tool)[65] [has] (high performances)[c,16]
$A_{1,42}$: (Designers)[c,62] [need an] (affordable software tool)[68]
$A_{1,43}$: (Designers)[c,62] [need] (step by step assistance)[69]
$A_{1,44}$: (Designers)[c,62] [need] (multifold data visualization)[72]
$A_{1,45}$: (Designers)[c,62] [need] (multi-channel data management)[74]
$A_{1,46}$: (Designers)[c,62] [need] (blended datasets)[76]
$A_{1,47}$: (Designers)[c,62] [need] (combined data)[78]
$A_{1,48}$: (Qualitative data)[c,20] [is included in] (data)[c,4]
$A_{1,49}$: (Quantitative data)[80] [is included in] (data)[c,4]
$A_{1,50}$: (Designers)[c,62] [need a] (permanently accessible software tool)[81]
$A_{1,51}$: (Designers)[c,62] [need a] (customized software tool)[82]
$A_{1,52}$: (Designer)[c,62] [need an] (intuitive data analytics tool)[83]
$A_{1,53}$: (Designers)[c,62] [need] (smart data analytics tool)[84]
$A_{1,54}$: (Designers)[c,62] [need] (case-based reasoning)[85]
$A_{1,55}$: (Semantic support)[c,66] [is provided to] (data)[c,4]
$A_{1,56}$: (Data collection practices)[c,21] [are multiple]
$A_{1,57}$: (Data collection practices)[c,21] [are heterogeneous]
$A_{1,58}$: (Sophisticated data management)[92] [merges] (data streams)[93]
$A_{1,59}$: (Sophisticated data management)[92] [facilitates] (data fusion)[94]
$A_{1,60}$: (Sophisticated data management)[92] [increases] (computational performances)[95]
$A_{1,61}$: (Sophisticated data management)[92] [improves] (usability)[97]
$A_{1,62}$: (Sophisticated data management)[92] [facilitates] (human interpretation)[98]
$A_{1,63}$: (Designers)[c,62] [need] (smart semantics)[99]

$A_{1,64}$: (Designers)[c,62] [need] (procedural reasoning)[101]
$A_{1,65}$: (Smart semantics)[99] [extracts] (MoL data meaning)[102]
$A_{1,66}$: (Smart semantics)[99] [extracts] (PLC data meaning)[110]
$A_{1,67}$: (Procedural reasoning)[101] [extracts] (MoL data meaning)[102]
$A_{1,68}$: (Procedural reasoning)[101] [extracts] (PLC data meaning)[110]
$A_{1,69}$: (Smart semantics)[99] [uses] (system intellect)[103]
$A_{1,70}$: (System intellect)[103] [is provided by] (AI)[104]
$A_{1,71}$: (System intellect)[103] [is provided by] (system learning mechanisms)[105]
$A_{1,72}$: (System intellect)[103] [is provided by] (context information processing)[106]
$A_{1,73}$: (System intellect)[103] [is provided by] (situation awareness)[107]
$A_{1,74}$: (System intellect)[103] [is provided by] (strategy development)[108]
$A_{1,75}$: (System intellect)[103] [is provided by] (system adaptation capabilities)[109]
$A_{1,76}$: (Smart semantics)[99] [extracts] (data meaning)[c,23]
$A_{1,77}$: (Procedural reasoning)[101] [extracts] (data meaning)[c,23]
$A_{1,78}$: (Designers)[c,62] [use] (data analytics tools)[c,1]
$A_{2,1}$: (Big data) [c,29] [is characterized by] its (volume)[c,30]
$A_{2,2}$: (Volume)[c,30] [refers to] (huge data amount)[3]
$A_{2,3}$: (Velocity)[c,31] [refers to] (data creation speed)[5]
$A_{2,4}$: (Velocity)[c,31] [refers to] (data stream speed)[6]
$A_{2,5}$: (Velocity)[c,31] [refers to] (data aggregation speed)[7]
$A_{2,6}$: (Velocity)[c,31] [refers to] (data movement speed)[8]
$A_{2,7}$: (Variety)[c,32] [refers to] (various data types)[142]
$A_{2,8}$: (Structured data)[c,34] [is] a (data type)[c,33]
$A_{2,9}$: (Semi-structured data)[c,35] [is] a (data type)[c,33]
$A_{2,10}$: (Unstructured data)[c,36] [is] a (data type)[c,33]
$A_{2,11}$: (Veracity)[c,37] [refers to] (data messiness)[15]
$A_{2,12}$: (Veracity)[c,37] [refers to] (data trustworthiness)[16]
$A_{2,13}$: (Variety)[c,32] [causes] (data messiness)[15]
$A_{2,14}$: (Volume)[c,30] [causes] (data messiness)[15]
$A_{2,15}$: (Value)[c,38] [refers to] (data meaning)[c,23]
$A_{2,16}$: (Big data analytics)[c,39] [examines] (big data) [c,29]
$A_{2,17}$: (Big data analytics)[c,39] [processes] (big data) [c,29]
$A_{2,18}$: (Big data analytics)[c,39] [reveals] (hidden patterns)[20]
$A_{2,19}$: (Big data analytics)[c,39] [identifies] (relationships)[21]
$A_{2,20}$: (Big data analytics)[c,39] [exposes] the (application context)[22]
$A_{2,21}$: (Advanced big data analytics)[23] [enables] (intelligent computerized solutions)[24]
$A_{2,22}$: (Advanced big data analytics)[23] [uses] (sentiment analysis)[25]
$A_{2,23}$: (Sentiment analysis)[25] [aims to improve] (products)[c,3]
$A_{2,24}$: (Sentiment analysis)[25] [aims to improve] (services)[27]
$A_{2,25}$: (Sentiment analysis)[25] [identifies] (user's opinion)[28]
$A_{2,26}$: (User's opinion)[28] [includes] their (evaluations)[29]
$A_{2,27}$: (User's opinion)[28] [includes] their (affective state)[30]
$A_{2,28}$: (Organizations)[c,40] [benefit from] (big data)[c,29]
$A_{2,29}$: (Organizations)[c,40] [benefit from] (cloud computing services)[32]
$A_{2,30}$: (Cloud computing services)[32] [store] (big data)[c,29]
$A_{2,31}$: (Could computing services)[32] [process] (big data analysis requirements)[33]

$A_{2,32}$: (Data)[c,4] [is accumulated from] (diverse data sources)[c,58]
$A_{2,33}$: (Diverse data sources)[c,58] [complicate] (online data processing)[36]
$A_{2,34}$: (Diverse data types)[147] [complicate] (online data processing)[36]
$A_{2,35}$: (Online data processing)[36] [synchronizes] (signal inputs)[37]
$A_{2,36}$: (Online data processing)[36] [analyzes] (data types)[c,33]
$A_{2,37}$: (Diverse data sources)[c,58] [produce] (variable signal inputs)[146]
$A_{2,38}$: (Real life applications)[38] [need] (high-speed storage)[39]
$A_{2,39}$: (Real life applications)[38] [need] (high-speed data processing)[40]
$A_{2,40}$: (Online adaptation)[44] [incorporates] (contextual elements)[45]
$A_{2,41}$: (Online adaptation)[44] [incorporates] (user-specific elements)[46]
$A_{2,42}$: (Contextual elements)[45] [are incorporated in] (design)[47]
$A_{2,43}$: (User-specific elements)[46] [are incorporated in] (design)[47]
$A_{2,44}$: (Contextual elements)[45] [are incorporated in] (decision-making mechanism)[48]
$A_{2,45}$: (User-specific elements)[46] [are incorporated in] (decision-making mechanism)[48]
$A_{2,46}$: (Big data analytics techniques)[43] [include] (computational techniques)[c,41]
$A_{2,47}$: (Big data analytics techniques)[43] [include] (ML techniques)[50]
$A_{2,48}$: (ML approaches)[c,59] [are used for] (patterns modeling)[52]
$A_{2,49}$: (ML approaches)[c,59] [are used for] (correlations modeling)[53]
$A_{2,50}$: (Patterns modeling)[52] [helps discovering] (relationships)[21]
$A_{2,51}$: (Correlations modeling)[53] [helps discovering] (relationships)[21]
$A_{2,52}$: (Patterns modeling)[52] [helps making] (predictions)[c,65]
$A_{2,53}$: (Correlations modeling)[53] [helps making] (predictions)[c,65]
$A_{2,54}$: (Predictions) [c,65] [are based on] (unseen data)[55]
$A_{2,55}$: (Predictions) [c,65] [are based on] (events) [c,79]
$A_{2,56}$: (Supervised learning)[57] [belongs to] (ML approaches)[c,59]
$A_{2,57}$: (Supervised learning)[57] [is from] (labelled data)[58]
$A_{2,58}$: (Unsupervised learning)[59] [belongs to] (ML approaches)[c,59]
$A_{2,59}$: (Unsupervised learning)[59] [discovers] (hidden patterns)[20]
$A_{2,60}$: (Reinforcement learning)[60] [belongs to] (ML approaches)[c,59]
$A_{2,61}$: (Reinforcement learning)[60] [is] (goal oriented learning)[61]
$A_{2,62}$: (Goal oriented learning)[61] [happens in] (dynamic situations)[62]
$A_{2,63}$: (Regression techniques)[c,42] [belongs to] (ML approaches)[c,59]
$A_{2,64}$: (Clustering approaches)[c,43] [belongs to] (ML approaches)[c,59]
$A_{2,65}$: (Density estimation methods)[65] [belongs to] (ML approaches)[c,59]
$A_{2,66}$: (Dimensionality reduction approaches)[66] [belongs to] (ML approaches)[c,59]
$A_{2,67}$: (CI)[c,60] [belongs to] (ML approaches)[c,59]
$A_{2,68}$: (Algorithms)[c,44] [are devised within] (CI)[c,60]
$A_{2,69}$: (Algorithms)[c,44] [imitate] (human information processing mechanisms)[69]
$A_{2,70}$: (Algorithms)[c,44] [imitate] (human information reasoning mechanisms)[70]
$A_{2,71}$: (Algorithms)[c,44] [aim at processing] (complex data sources)[143]
$A_{2,72}$: (Algorithms)[c,44] [aim at processing] (uncertain data sources)[144]
$A_{2,73}$: (CI techniques)[71] [addresses] (complex real-world data driven problems)[145]
$A_{2,74}$: (Mathematical modeling)[73] [does not solve] (complex real-world data driven problems)[145]
$A_{2,75}$: (Traditional modeling)[74] [does not solve] (complex real-world data driven problems)[145]

$A_{2,76}$: (Complex real-world data driven problems)[145] [have] (highly complex processes)[75]
$A_{2,77}$: (Complex real-world data driven problems)[145] [have] (highly uncertain processes)[76]
$A_{2,78}$: (Complex real-world data driven problems)[145] [have] (highly stochastic processes)[77]
$A_{2,79}$: (FL)[78] [belongs to] (CI approaches)[81]
$A_{2,80}$: (EA)[79] [belongs to] (CI approaches)[81]
$A_{2,81}$: (ANN)[80] [belongs to] (CI approaches)[81]
$A_{2,82}$: (FL)[78] [handles] (real-world data driven problems)[72]
$A_{2,83}$: (EA)[79] [handles] (real-world data driven problems)[72]
$A_{2,84}$: (ANN)[80] [handles] (real-world data driven problems)[72]
$A_{2,85}$: (FL)[78] [is] a (methodology)[82]
$A_{2,86}$: (FL)[78] [deals with] (imprecise data)[83]
$A_{2,87}$: (FL)[78] [deals with] (uncertain data)[84]
$A_{2,88}$: (FL)[78] [models] (qualitative data)[c,20]
$A_{2,89}$: (FL)[78] [models] (adaptive control)[86]
$A_{2,90}$: (FL)[78] [models using] (linguistic quantifiers)[87]
$A_{2,91}$: (Linguistic quantifiers)[87] [are called] (fuzzy sets)[88]
$A_{2,92}$: (Fuzzy sets)[88] [help representing] (uncertain real-world)[89]
$A_{2,93}$: (Fuzzy sets)[88] [help representing] (uncertain data)[84]
$A_{2,94}$: (Fuzzy sets)[88] [help representing] (uncertain user defined concepts)[90]
$A_{2,95}$: (Fuzzy sets)[88] [help representing] (human interpretable fuzzy rules)[91]
$A_{2,96}$: (Uncertain real-world)[89] [is used for] (inference)[92]
$A_{2,97}$: (Uncertain data)[84] [is used for] (inference)[92]
$A_{2,98}$: (Uncertain user defined concepts)[90] [are used for] (inference)[92]
$A_{2,99}$: (Human interpretable fuzzy rules)[91] [are used for] (inference)[92]
$A_{2,100}$: (Uncertain real-word)[89] [is used for] (decision-making)[c,61]
$A_{2,101}$: (Uncertain data)[84] [is used for] (decision-making)[c,61]
$A_{2,102}$: (Uncertain user defined concepts)[90] [are used for] (decision-making)[c,61]
$A_{2,103}$: (Human interpretable fuzzy rules)[91] [are used for] (decision-making)[c,61]
$A_{2,104}$: (EA)[79] [is based on] (genetic algorithms)[94]
$A_{2,105}$: (EA)[79] [is based on] (genetic programming)[95]
$A_{2,106}$: (EA)[79] [is based on] (swarm intelligence optimization algorithms)[96]
$A_{2,107}$: (EA)[79] [optimizes] (complex real-world systems)[97]
$A_{2,108}$: (EA)[79] [optimizes] (complex real-world processes)[98]
$A_{2,109}$: (ANN)[80] [enables] (feature extraction)[99]
$A_{2,110}$: (ANN)[80] [learns from] (experiential data)[100]
$A_{2,111}$: (CI techniques combination)[101] [extracts] (insights)[c,45]
$A_{2,112}$: (Insights)[c,45] [are extracted from] (data)[c,4]
$A_{2,113}$: (CI techniques combination)[101] [extracts] (data meaning)[c,23]
$A_{2,114}$: (CI techniques combination)[101] [offers] (integrated solutions)[103]
$A_{2,115}$: (CI techniques combination)[101] [provides] (effective multipurpose intelligent data analysis)[105]
$A_{2,116}$: (CI techniques combination)[101] [provides] (effective decision-making)[106]
$A_{2,117}$: (Big data analytics)[c,39] [has] (needs)[c,22]

$A_{2,118}$: (Important features identification)[108] [is] a (need)[c,22]
$A_{2,119}$: (Important features)[109] [are extracted from] (data)[c,4]
$A_{2,120}$: (Data)[c,4] [affects] (outputs)[c,46]
$A_{2,121}$: (Spatial co-relations identification)[111] [is] a (need)[c,22]
$A_{2,122}$: (Spatial co-relations)[112] [are between] a (input variables)[113]
$A_{2,123}$: (Spatial co-relations)[112] [are identified at] a (given time)[114]
$A_{2,124}$: (Temporal co-relations identification)[115] [is] a (need)[c,22]
$A_{2,125}$: (Temporal co-relations)[116] [are between] (input parameters)[117]
$A_{2,126}$: (Input parameters)[117] [change] (overtime)[118]
$A_{2,127}$: (DL approaches)[119] [model] (Large-scale data)[120]
$A_{2,128}$: (DL approaches)[119] [model] (spatial correlations)[121]
$A_{2,129}$: (DL approaches)[119] [model] (temporal correlations)[122]
$A_{2,130}$: (DL approaches)[119] [uses] (ANN)[80]
$A_{2,131}$: (DL approaches)[119] [uses] (multiple hidden layers)[123]
$A_{2,132}$: (Feature learning methods)[124] [are based on] (supervised approaches)[125]
$A_{2,133}$: (DNNs)[126] [are] (supervised approaches)[125]
$A_{2,134}$: (CNNs)[127] [are] (supervised approaches)[125]
$A_{2,135}$: (Recurrent NNs)[128] [are] (supervised approaches)[125]
$A_{2,136}$: (DL techniques)[129] [analyses] (big data)[c,29]
$A_{2,137}$: (DL techniques)[129] [processes] (big data)[c,29]
$A_{2,138}$: (DNN)[126] [perform] (pattern recognition)[130]
$A_{2,139}$: (DNN)[126] [are] (computationally expensive)[131]
$A_{2,140}$: (DNN)[126] [require] (training time)[132]
$A_{2,141}$: (FL)[78] [relies on] (natural language fuzzy rules)[133]
$A_{2,142}$: (FL)[78] [visualize] (hidden relations)[134]
$A_{2,143}$: (FL)[78] [visualize] (hidden patterns)[20]
$A_{2,144}$: (FL)[78] [helps creating] (user-friendly systems)[136]
$A_{2,145}$: (EA)[79] [are] (excellent data analysis tools)[137]
$A_{2,146}$: (EA)[79] [deal with] (high dimensionality)[138]
$A_{2,147}$: (EA)[79] [deal with] (sparseness)[139]
$A_{2,148}$: (EA)[79] [perform] (data clustering)[140]
$A_{2,149}$: (EA)[79] [perform] (feature selection)[141]
$A_{2,150}$: (Big data)[c,29] [is characterized by] its (velocity)[c,31]
$A_{2,151}$: (Big data)[c,29] [is characterized by] its (variety)[c,32]
$A_{2,152}$: (Big data)[c,29] [is characterized by] its (veracity)[c,37]
$A_{2,153}$: (Big data)[c,29] [is characterized by] its (value)[c,38]
$A_{3,1}$: (Data analytics)[c,48] [includes] (data collection)[c,21]
$A_{3,2}$: (Data analytics)[c,48] [includes] (data organization)[3]
$A_{3,3}$: (Data analytics)[c,48] [analyzes] (large datasets)[117]
$A_{3,4}$: (Big data analytics)[c,39] [processes] (datasets)[c,19]
$A_{3,5}$: (Big data analytics)[c,39] [analyzes huge] (datasets)[c,19]
$A_{3,6}$: (Big data)[c,29] [cannot be processed using] (traditional data analytics techniques)[119]
$A_{3,7}$: (Digitalization)[6] [increases] (data quality)[7]
$A_{3,8}$: (Digitalization)[6] [increases] (data variety)[8]
$A_{3,9}$: (Data analytics)[c,48] [examines] (raw data)[120]
$A_{3,10}$: (Data analytics)[c,48] [identifies] (data behavior)[10]

$A_{3,11}$: (Data analytics)^{c,48} [analyzes] (data behavior)^{10}
$A_{3,12}$: (Data analytics)^{c,48} [identifies] (patterns)^{c,47}
$A_{3,13}$: (Data analytics)^{c,48} [analyzes] (patterns)^{c,47}
$A_{3,14}$: (Patterns)^{c,47} [are identified using] (qualitative techniques)^{12}
$A_{3,15}$: (Patterns)^{c,47} [are identified using] (quantitative techniques)^{13}
$A_{3,16}$: (Patterns)^{c,47} [are analyzed using] (qualitative techniques)^{12}
$A_{3,17}$: (Patterns)^{c,47} [are analyzed using] (quantitative techniques)^{13}
$A_{3,18}$: (Data behavior)^{10} [is identified using] (qualitative techniques)^{12}
$A_{3,19}$: (Data behavior)^{10} [is identified using] (quantitative techniques)^{13}
$A_{3,20}$: (Data behavior)^{10} [is analyzed using] (qualitative techniques)^{12}
$A_{3,21}$: (Data behavior)^{10} [is analyzed using] (quantitative techniques)^{13}
$A_{3,22}$: (Data analytics)^{c,48} [aims at deriving] (actionable insights)^{121}
$A_{3,23}$: (Data)^{c,4} [reveals] (customer preferences)^{15}
$A_{3,24}$: (Data analytics tools)^{c,1} [are profitable for] (organizations)^{c,40}
$A_{3,25}$: (Data analytics tools)^{c,1} [are productive for] (organizations)^{c,40}
$A_{3,26}$: (Data analytics techniques)^{c,57} [are profitable for] (organizations)^{c,40}
$A_{3,27}$: (Data analytics techniques)^{c,57} [are productive for] (organizations)^{c,40}
$A_{3,28}$: (Descriptive analytics)^{18} [belongs to] (analytical techniques)^{19}
$A_{3,29}$: (Predictive analytics)^{20} [belongs to] (analytical techniques)^{19}
$A_{3,30}$: (Prescriptive analytics)^{21} [belongs to] (analytical techniques)^{19}
$A_{3,31}$: (Descriptive analytics)^{18} [is based on] (historic data)^{22}
$A_{3,32}$: (Descriptive analytics)^{18} [develops] (new insights)^{128}
$A_{3,33}$: (Descriptive analytics)^{18} [uses] (probability analysis)^{23}
$A_{3,34}$: (Descriptive analytics)^{18} [uses] (trending)^{24}
$A_{3,35}$: (Descriptive analytics)^{18} [uses] (classified data association development)^{129}
$A_{3,36}$: (Descriptive analytics)^{18} [uses] (categorized data association development)^{131}
$A_{3,37}$: (Predictive analytics)^{20} [predicts] (future outcomes)^{132}
$A_{3,38}$: (Predictive analytics)^{20} [informs about] (happenings)^{28}
$A_{3,39}$: (Predictive analytics)^{20} [informs about] (actions)^{c,63}
$A_{3,40}$: (Predictive analytics)^{20} [provides] (happenings time frame)^{30}
$A_{3,41}$: (Predictive analytics)^{20} [derives] (the best possible outcome)^{135}
$A_{3,42}$: (Prescriptive analytics)^{21} [analyzes] (the possible outcomes)^{136}
$A_{3,43}$: (Prescriptive analytics)^{21} [is flexible]
$A_{3,44}$: (Prescriptive analytics)^{21} [improves with] (experience)^{31}
$A_{3,45}$: (Prescriptive analytics)^{21} [is applicable to] (specific domains)^{32}
$A_{3,46}$: (Prescriptive analytics)^{21} [is applicable to] (specific processes)^{33}
$A_{3,47}$: (Problem identification)^{36} [is used in implementing an] (analytics project)^{35}
$A_{3,48}$: (Gathering required data)^{37} [is used in implementing an] (analytics project)^{35}
$A_{3,49}$: (Data pre-processing)^{38} [is used in implementing an] (analytics project)^{35}
$A_{3,50}$: (Performing analytics over data)^{39} [is used in implementing an] (analytics project)^{35}
$A_{3,51}$: (Data visualization)^{c,18} [is used in implementing an] (analytics project)^{35}
$A_{3,52}$: (Gathering required data)^{37} [is useful for] the (analysis)^{c,49}
$A_{3,53}$: (Gathering required data)^{37} [is done through] (various data sources)^{c,58}
$A_{3,54}$: (Data attributes)^{43} of (datasets)^{c,19} [are defined]
$A_{3,55}$: (Problem identification)^{36} [determines] (data attributes)^{43}

$A_{3,56}$: (Data pre-processing)[38] [translates] (data)[c,4]
$A_{3,57}$: (Data)[c,4] [is translated into] (fixed data format)[144]
$A_{3,58}$: (Fixed data format)[144] [is provided to] (algorithms)[c,44]
$A_{3,59}$: Fixed (data format)[144] [is provided to] (data analytics tools)[c,1]
$A_{3,60}$: (Performing analytics over data)[39] [is executed using] (algorithmic concepts)[46]
$A_{3,61}$: (Regression)[c,42] [is] an (algorithmic concept)[46]
$A_{3,62}$: (Classification)[48] [is] an (algorithmic concept)[46]
$A_{3,63}$: (Clustering)[c,43] [is] an (algorithmic concept)[46]
$A_{3,64}$: (Model-based recommendation)[50] [is] an (algorithmic concept)[46]
$A_{3,65}$: (Data visualization)[c,18] [displays] (data analytics outputs)[51]
$A_{3,66}$: (Data visualization)[c,18] [interactively represents] (insights)[c,45]
$A_{3,67}$: (Advanced data analytics techniques)[148] [include] (behavioral analytics)[52]
$A_{3,68}$: (Advanced data analytics techniques)[148] [include] (social media analytics)[53]
$A_{3,69}$: (Advanced data analytics techniques)[148] [include] (time series analysis)[54]
$A_{3,70}$: (Data sources)[c,52] [include] (videos)[55]
$A_{3,71}$: (Data sources)[c,52] [include] (audio)[56]
$A_{3,72}$: (Data sources)[c,52] [include] (digital images)[57]
$A_{3,73}$: (Data sources)[c,52] [include] (sensors)[58]
$A_{3,74}$: (Data sources)[c,52] [include] (log files)[59]
$A_{3,75}$: (Data sources)[c,52] [include] (transactional applications)[60]
$A_{3,76}$: (Data sources)[c,52] [include] (web)[61]
$A_{3,77}$: (Data sources)[c,52] [include] (social media)[62]
$A_{3,78}$: (Big data)[c,29] [is generated in] (real time)[63]
$A_{3,79}$: (Big data)[c,29] [is generated in] (large scale)[64]
$A_{3,84}$: (Volume)[c,30] [refers to] (dataset size)[69]
$A_{3,85}$: (Variety)[c,32] [refers to] (data types)[c,33]
$A_{3,86}$: (Variety)[c,32] [increases] (data complexity)[71]
$A_{3,87}$: (Velocity)[c,31] [refers to] (data generation speed)[72]
$A_{3,88}$: (Velocity)[c,31] [refers to] (data delivery speed)[73]
$A_{3,89}$: (Veracity)[c,37] [refers to] (data quality)[7]
$A_{3,90}$: (Veracity)[c,37] [refers to] (data availability)[74]
$A_{3,91}$: (Structured data)[c,34] [is arranged in] a (definite pattern)[153]
$A_{3,92}$: (Relational database management systems)[76] [build] (structured data)[c,34]
$A_{3,93}$: (Semi-structured data)[c,35] [has no] (pre-defined structure)[156]
$A_{3,94}$: (Semi-structured data)[c,35] [has] (different structures)[157]
$A_{3,95}$: (Semi-structured data)[c,35] [includes] (scientific data)[79]
$A_{3,96}$: (Semi-structured data)[c,35] [includes] (bibliographic data)[80]
$A_{3,97}$: (Graph data structures)[81] [holds] (semi-structured data)[c,35]
$A_{3,98}$: (Unstructured data)[c,36] [has no] (standard structure)[158]
$A_{3,99}$: (Videos)[55] [are] (unstructured data)[c,36]
$A_{3,100}$: (Digital images)[57] [are] (unstructured data)[c,36]
$A_{3,101}$: (Documents)[83] [are] (unstructured data)[c,36]
$A_{3,102}$: (Multi-structured data)[84] [is] (data mix)[85]
$A_{3,103}$: (Structured data)[c,34] [is part of] (data mix)[85]
$A_{3,104}$: (Semi-structured data)[c,35] [is part] (data mix)[85]
$A_{3,105}$: (Operating system level logs)[86] [are] (multi-structured data)[84]

$A_{3,106}$: (Operating system level logs)[86] [are used for] (performance)[c,17]
$A_{3,107}$: (Operating system level logs)[86] [are used for] (monitoring)[88]
$A_{3,108}$: (Unstructured data)[c,36] [is part of] (data mix)[85]
$A_{3,109}$: (Big data)[c,29] [has] (challenges)[89]
$A_{3,110}$: The (challenges)[89] [are grouped into] (dimensions)[90]
$A_{3,111}$: (Data challenge)[91] [is] a (dimensions)[90]
$A_{3,112}$: (Process challenge)[92] [is] a (dimensions)[90]
$A_{3,113}$: (Management challenge)[93] [is] a (dimensions)[90]
$A_{3,114}$: [Handling] (huge information)[c,70] [is] a (data challenge)[91]
$A_{3,115}$: [Handling] (multiple data types)[161] [is] a (data challenge)[91]
$A_{3,116}$: [Handling] (multiple data sources)[162] [is] a (data challenge)[91]
$A_{3,117}$: [Handling] (multiple data formats)[163] [is] a (data challenge)[91]
$A_{3,118}$: [Coping with] (uncertainty)[95] [is] (data challenge)[91]
$A_{3,119}$: [Coping with] (imprecision)[96] [is] a (data challenge)[91]
$A_{3,120}$: [Coping with] (missing values)[164] [is] a (data challenge)[91]
$A_{3,121}$: [Coping with] (missing statements)[165] [is] a (data challenge)[91]
$A_{3,122}$: [Coping with] (untruths)[99] [is] a (data challenge)[91]
$A_{3,123}$: (Data discovery)[100] [is a] (huge data challenge)[166]
$A_{3,124}$: [Finding] (high data quality)[167] [is] a (data challenge)[91]
$A_{3,125}$: [Determining] (data relevance)[101] [is] a (data challenge)[91]
$A_{3,126}$: (Data comprehensiveness)[102] [is] a (data challenge)[91]
$A_{3,127}$: [Processing] (huge datasets)[168] [is] a (process challenge)[92]
$A_{3,128}$: [Extracting] (significant information)[c,75] [is] a (process challenge)[92]
$A_{3,129}$: (Data collection)[c,21] [is] a (process challenge)[92]
$A_{3,130}$: [Understanding] (outputs)[c,46] [is] a (process challenge)[92]
$A_{3,131}$: (Data alignment)[104] [is] a (process challenge)[92]
$A_{3,132}$: (Data transformation)[105] [is] a (process challenge)[92]
$A_{3,133}$: (Data modeling)[106] [is] a (process challenge)[92]
$A_{3,134}$: (Results visualization)[107] [is] a (process challenge)[92]
$A_{3,135}$: (Results sharing)[108] [is] a (process challenge)[92]
$A_{3,136}$: (Data privacy)[109] [is] a (management challenge)[93]
$A_{3,137}$: (Data security)[110] [is] a (management challenge)[93]
$A_{3,138}$: (Data governance)[111] [is] a (management challenge)[93]
$A_{3,139}$: [Correctly using] (data)[c,4] [is] a (management challenge)[93]
$A_{3,140}$: (Data)[c,4] [is mostly unstructured]
$A_{3,141}$: (Traditional data processing systems)[170] [cannot process] (huge datasets)[168]
$A_{4,1}$: (Design problems)[1] concerning (products)[c,3] [are not clearly identifiable].
$A_{4,2}$: (A product)[c,3] [is characterized by] (functional requirements)[3].
$A_{4,3}$: (Functional requirements)[3] [are not exhaustively specified].
$A_{4,4}$: (Product information)[53] [is incomplete].
$A_{4,5}$: (Product information)[53] [is imperfect].
$A_{4,6}$: (Product knowledge)[54] [is incomplete].
$A_{4,7}$: (Product knowledge)[54] [is imperfect].
$A_{4,8}$: (The courses of action)[8] [are unclear for] (decision-makers)[c,62].
$A_{4,9}$: (Creativity)[12] [appears as a] (purposive novelty)[55].
$A_{4,10}$: (A difficult design problem)[56] [admits no] (obvious solution)[57].

$A_{4,11}$: (A creative solution)[58] [solves] (difficult design problem)[56].
$A_{4,12}$: (Knowledge)[c,2] [is acquired through] (the course of action)[8].
$A_{4,13}$: (Knowledge)[c,2] [is inherent in] (the system)[c,55] .
$A_{4,14}$: (Knowledge)[c,2] [is used for] (decision-making) [c,61].
$A_{4,15}$: (Information)[c,10] [is a subset of] (knowledge)[c,2].
$A_{4,16}$: (Data)[c,4] [is a precursor of] (product information)[53].
$A_{4,17}$: (Knowledge-based systems)[59] [are used to support] (high-level decision-making)[60].
$A_{4,18}$: (Design process)[19] [fulfils] (functional requirements)[3] .
$A_{4,19}$: (Knowledge-based systems)[59] [are used to enhance] (decision-making) [c,61] .
$A_{4,20}$: (Information)[c,10] [is crucial for] (operations)[17].
$A_{4,21}$: (Decision quality)[20] [is linearly proportional with the] (level of proactivity)[21].
$A_{4,22}$: (6R)[22] [is a] (decision-making process)[64].
$A_{4,23}$: (6R)[22] [presents] (relevant information)[c,75].
$A_{4,24}$: (6R)[22] [is needed by] (decision-makers)[c,62].
$A_{4,25}$: (6R)[22] [is relevant for] (decision-making)[c,61].
$A_{4,26}$: (6R)[22] [inquires] (observation)[23].
$A_{4,27}$: (6R)[22] [inquires] (analysis)[c,49].
$A_{4,28}$: (6R)[22] [inquires] (diagnostic)[25].
$A_{4,29}$: (6R)[22] [inquires] (prediction)[c,65].
$A_{4,30}$: (6R)[22] [inquires] (prescription)[27].
$A_{4,31}$: (6R)[22] [inquires] (semantic interpretation)[c,66].
$A_{4,32}$: (Observation)[23] [is realized before] (analysis)[c,49].
$A_{4,33}$: (Analysis)[c,49] [is realized before] (diagnostic)[25].
$A_{4,34}$: (Diagnostic)[25] [is realized before] (prediction)[c,65].
$A_{4,35}$: (Prediction)[c,65] [is realized before] (prescription)[27].
$A_{4,36}$: (Prescription)[27] [is realized before] (semantic interpretation)[c,66].
$A_{4,37}$: (Decision-making)[c,61] [is fed by] (data)[c,4].
$A_{4,38}$: (Robust decision-making)[64] [quickly identify] (key influencing events)[29].
$A_{4,39}$: (Robust decision-making)[64] [requires to make] (appropriate inquiries)[30].
$A_{4,40}$: (Appropriate inquires)[30] [concerns] (key influencing events)[29].
$A_{4,41}$: (Context-driven decision-making)[70] [is based on] (dynamically integrated knowledge)[71].
$A_{4,42}$: (Dynamically integrated knowledge)[71] [is relevant to] (mission)[31].
$A_{4,43}$: (Dynamically integrated knowledge)[71] [is relevant to] (environment)[32].
$A_{4,44}$: (Dynamically integrated knowledge)[71] [is relevant to] (assets)[33].
$A_{4,45}$: (Dynamically integrated knowledge)[71] [is relevant to] (tasks)[c,80].
$A_{4,46}$: (Dynamically integrated knowledge)[71] [is relevant to] (threats)[35].
$A_{4,47}$: (Dynamically integrated knowledge)[71] [is informed by] (up-to-date data sources)[36].
$A_{4,48}$: (Dynamically integrated knowledge)[71] [is congruent with] (objectives)[c,73].
$A_{4,49}$: (Dynamically integrated knowledge)[71] [is congruent to] (workloads)[38].
$A_{4,50}$: (Dynamically integrated knowledge)[71] [is congruent to] (roles)[c,81].
$A_{4,51}$: (Dynamically integrated knowledge)[71] [is congruent to] (constraints)[4].
$A_{4,52}$: (Unmanned systems)[44] [are combined with] (smart reasoning technologies)[45].
$A_{4,53}$: (Real-time data)[73] [is difficult to present to] (decision-makers)[c,62].

$A_{4,54}$: (Relevant information)[c,75] [is difficult to extract from] (real-time data)[73].
$A_{4,55}$: (Huge information)[c,70] [overwhelms] (decision-makers)[c,62].
$A_{4,56}$: (Decision-makers)[c,62] [infer] (courses of action)[8].
$A_{4,57}$: (Decision-makers)[c,62] [reconstruct] (courses of action)[8].
$A_{4,58}$: (Humans)[48] [are] (decision-makers)[c,62].
$A_{4,59}$: (Autonomous agents)[49] [are] (decision-makers)[c,62].
$A_{4,60}$: (Action)[c,63] [is executed using] (assets)[33].
$A_{4,61}$: (Action)[c,63] [is executed using] (resources)[50].
$A_{4,62}$: (Decision-support systems)[76] (exploit) (relevant information)[c,75].
$A_{4,63}$: (New context)[77] [is hard to classify correctly].
$A_{4,64}$: (Decision-makers)[c,62] [specify] (constraints)[4].
$A_{4,65}$: (Context)[c,67] [is diverse].
$A_{4,66}$: (Context)[c,67] [facilitates to determine] (knowledge)[c,2].
$A_{4,67}$: (Context)[c,67] [is important for using] (knowledge-based systems)[59].
$A_{5,1}$: (HSI)[1] [is influenced by] (the levels of interaction)[2]
$A_{5,2}$: (HSI)[1] [is influenced by] (the domains of interaction)[3]
$A_{5,3}$: (HSI)[1] [is influenced by] (the contexts of interaction)[4]
$A_{5,4}$: (HSI)[1] [is influenced by] (the modalities of interaction)[5]
$A_{5,5}$: (Physical level of interaction)[41] [is a] (level of interaction)[2]
$A_{5,6}$: (Syntactic level of interaction)[42] [is a] (level of interaction)[2]
$A_{5,7}$: (Semantic level of interaction)[43] [is a] (level of interaction)[2]
$A_{5,8}$: (Pragmatic level of interaction)[44] [is a] (level of interaction)[2]
$A_{5,9}$: (Apobetic level of interaction)[45] [is a] (level of interaction)[2]
$A_{5,10}$: (Perceptive domain of interaction)[46] [is a] (domain of interaction)[3]
$A_{5,11}$: (Cognitive domain of interaction)[47] [is a] (domain of interaction)[3]
$A_{5,12}$: (Motor domain of interaction)[48] [is a] (domain of interaction)[3]
$A_{5,13}$: (Emotional domain of interaction)[49] [is a] (domain of interaction)[3]
$A_{5,14}$: (HHI)[6] [is complemented by] (HTI)[7]
$A_{5,15}$: (HHI)[6] [is complemented by] (HIS)[1]
$A_{5,16}$: (HMI)[8] [belongs to] (HIS)[1]
$A_{5,17}$: (HCI)[9] [belongs to] (HIS)[1]
$A_{5,18}$: (Human role)[10] [is changing with] (the system)[11]
$A_{5,20}$: (Interaction)[18] [depends on the] (context)[c,67]
$A_{5,21}$: (The intention)[20] [is formed within] (the HCI)[9]
$A_{5,22}$: (An action)[c,63] [is selected within] (the HCI)[9]
$A_{5,23}$: (An action)[c,63] [is executed within] (the HCI)[9]
$A_{5,24}$: (The outcomes)[c,11] [are evaluated within] (the HCI)[9]
$A_{5,25}$: (Forming the intention)[20] [is a] (mental activity)[23]
$A_{5,26}$: (Selecting an action)[51] [is a] (mental activity)[23]
$A_{5,27}$: (Executing the action)[52] [is dominated by a] (physical activity)[53]
$A_{5,28}$: (Evaluating the outcomes)[22] [is a] (mental activity)[23]
$A_{5,29}$: (HMI)[8] [is presented by describing] (human factors)[54]
$A_{5,30}$: (HMI)[8] [is presented by describing] (machine factors)[55]
$A_{5,31}$: (HMI)[8] [is presented by describing] (interaction factors)[56]
$A_{5,32}$: (Human behaviour)[57] [is captured in] (levels)[26]
$A_{5,33}$: (Human behaviour)[57] [is organized in] (levels)[26]

$A_{5,34}$: (Skill)[27] [is a] (level)[26]
$A_{5,35}$: (Rule)[c,76] [is a] (level)[26]
$A_{5,36}$: (Knowledge)[c,2] [is a] (level)[26]
$A_{5,37}$: (CPSs)[17] [are] (smart systems)[58]
$A_{5,38}$: (CPSs)[17] [are] (adaptive systems)[59]
$A_{5,39}$: (CPSs)[17] [are] (decentralized systems)[60]
$A_{5,40}$: (CPSs)[17] [are] (distributed systems)[61]
$A_{5,41}$: (CPSs)[17] [are] (multi-scale systems)[62]
$A_{5,42}$: (CPSs)[17] [have] (varied components)[63]
$A_{5,43}$: (CPSs)[17] [have] (internal relations)[64]
$A_{5,44}$: (CPSs)[17] [have] (external relations)[65]
$A_{5,45}$: (Physical connectivity)[66] [makes it possible to implement] (syntactic interaction)[67]
$A_{5,46}$: (Semantic interpretation)[c,66] [is realized based on] (human mental processes)[69]
$A_{5,47}$: (Human mental process)[69] [consider] (the context)[c,67]
$A_{5,48}$: (Human mental process)[69] [interpret] (the objectives)[c,73]
$A_{5,49}$: (Semantic technologies)[70] [are needed between] (computational actors)[71]
$A_{5,50}$: (Ontologies)[38] [are] (semantic technologies)[70]
$A_{5,51}$: (Reasoning engines)[39] [are] (semantic technologies)[70]
$A_{5,52}$: (CPSs interactions)[72] [go beyond] (traditional HCI)[73]
$A_{5,53}$: (CPSs interactions)[72] [go beyond] (traditional HMI)[74]
$A_{5,54}$: (Actors)[37] [are multiples in] (CPSs interactions)[72]
$A_{5,55}$: (Internal interrelationships)[75] [are multiple in] (CPSs interactions)[72]
$A_{5,56}$: (External interrelationships)[75] [are multiple in] (CPSs interactions)[72]
$A_{5,57}$: (Actors)[37] [have] (diverse interactions)[77]

Derived postulates from the five component theories:

$P^{D}_{1,1}$: (Designers)[c,62] [want to have a] (complete software tool)[65]
$P^{D}_{1,2}$: (Designers)[c,62] [want to be advised in their] (choices)[118]
$P^{D}_{1,3}$: (Toolbox)[58] [should learn from its] (application)[c,26]
$P^{D}_{1,4}$: (Toolbox)[58] [should include] (case-based reasoning)[85]
$P^{D}_{1,5}$: (Data analytics tools)[c,1] [should be intuitive]
$P^{D}_{1,6}$: (Data analytics tools)[c,1] [should be smart]
$P^{D}_{1,7}$: (Designers)[c,62] [want to have] (semantic support)[c,66]
$P^{D}_{1,8}$: (Data analytics solutions)[120] [should be in harmony with] (data collection practices)[c,21]
$P^{D}_{1,9}$: (Data analytics solutions)[120] [should be in harmony with] (analytical needs)[90]
$P^{D}_{1,10}$: (Data analytics solutions)[120] [should cope with] (incomplete data)[123]
$P^{D}_{2,1}$: (Big data analytics methods)[41] [should extract] (interpretable knowledge)[42]
$P^{D}_{2,2}$: (Big data analytics techniques)[43] [should transparentize] (patterns)[c,47]
$P^{D}_{2,3}$: (Big data analytics techniques)[43] [should perform] (online adaptation)[44]
$P^{D}_{2,4}$: (Online adaptation)[44] [should be] user-friendly
$P^{D}_{2,5}$: (Online adaptation)[44] [should be] computationally feasible
$P^{D}_{2,6}$: (Integrated solutions)[103] [should be applied to] (online data processing)[36]
$P^{D}_{2,7}$: (Integrated solutions)[103] [should be applied to] (offline data processing)[104]
$P^{D}_{3,1}$: (Big data technologies)[113] [should be used in] a (secure way)[114]

$P^{D}_{3,2}$: (Data)[c,4] [must be formatted to be suitable for] (data mining)[115]
$P^{D}_{3,3}$: (Data)[c,4] [must be formatted to be suitable for] (subsequent analysis)[116]
$P^{D}_{4,1}$: (Constraints)[4] of (creative decision-making)[61] [are nebulous].
$P^{D}_{4,2}$: (Constraints)[4] of (creative decision-making)[61] [are ill-defined].
$P^{D}_{4,3}$: (Product information)[53] [is contradictory].
$P^{D}_{4,4}$: (Product knowledge)[54] [is contradictory].
$P^{D}_{4,5}$: (An action)[c,63] [must be monitored in terms of its] (impact)[10] .
$P^{D}_{4,6}$: (Decision-makers)[c,62] [have to execute] (design tasks)[c,64].
$P^{D}_{4,7}$: (Constraints)[4] [are to be considered to introduce a] (new product)[c,78].
$P^{D}_{4,8}$: (Functional requirements)[3] [are to be considered to solve] (design problems)[1].
$P^{D}_{4,9}$: (Proactive decision-making)[42] [should be] responsive.
$P^{D}_{4,10}$: (Proactive decision-making)[42] [should be] robust.
$P^{D}_{4,11}$: (Proactive decision-making)[42] [should be] innovative.
$P^{D}_{4,12}$: (Proactive decision-making)[42] [should be] flexible.
$P^{D}_{4,13}$: (Proactive decision-making)[42] [should anticipate] (context changes)[43]
$P^{D}_{4,14}$: (Knowledge-based systems)[59] [are reliable].
$P^{D}_{4,15}$: (Real-time data)[73] [is not directly accessed by] (humans)[48].
$P^{D}_{4,16}$: (Context)[c,67] [is poorly used in] (decision-making)[c,61].
$P^{D}_{4,17}$: (Data)[c,4] [is processed to obtain] (actionable decisions)[78].
$P^{D}_{5,1}$: (Awareness)[c,68] [is to be considered as] (intelligent systems)[78] [emerge]
$P^{D}_{5,2}$: (Proactivity)[16] [is to be considered as] (intelligent systems)[78] [emerge]
$P^{D}_{5,3}$: (Intelligence-based SHI)[79] [is to be considered as] (intelligent systems)[78] [emerge]
$P^{D}_{5,4}$: (HHI)[6] [is to be considered in designing] (CPSs)[17]
$P^{D}_{5,5}$: (HIS)[1] [is to be considered in designing] (CPSs)[17]
$P^{D}_{5,6}$: (SHI)[12] [is to be considered in designing] (CPSs)[17]
$P^{D}_{5,7}$: (SSI)[13] [is to be considered in designing] (CPSs)[17]

<u>Auxiliary postulates added:</u>

$P^{A}_{1,1}$: (Use data)[5] [belong to] (MoL data)[6]
$P^{A}_{1,2}$: (MoL data)[6] [are collected during] (MoL)[7]
$P^{A}_{1,3}$: (MoL)[7] [belongs to] (PLC)[8]
$P^{A}_{1,4}$: (PLC)[8] [includes] (EoL)[13]
$P^{A}_{1,5}$: (PLC)[8] [includes] (BoL)[14]
$P^{A}_{1,6}$: (PLC data)[15] [include] (EoL data)[16]
$P^{A}_{1,7}$: (PLC data)[15] [include] (BoL data)[17]
$P^{A}_{1,8}$: (PLC data)[15] [is collected during] (PLC)[8]
$P^{A}_{1,9}$: (EoL data)[16] [is collected during] (EoL)[13]
$P^{A}_{1,10}$: (BoL data)[17] [is collected during] (BoL)[14]
$P^{A}_{1,11}$: (PLC data)[15] [are] (data)[c,4]
$P^{A}_{1,12}$: (Work data)[10] [belongs to] (MoL data)[6]
$P^{A}_{1,13}$: (Failure data)[11] [belongs to] (MoL data)[6]
$P^{A}_{1,14}$: (Product developer)[18] [is a] (developer)[19]
$P^{A}_{1,15}$: (Product enhancement)[22] [concerns] (products)[c,3]
$P^{A}_{1,16}$: (Data analytics method)[c,5] [is a] (method)[c,6]
$P^{A}_{1,17}$: (Smart product)[24] [is a] (product)[c,3]

$P^A_{1,18}$: (Self-learning capability)[25] [is a] (capability)[28]
$P^A_{1,19}$: (Self-adaptation capability)[26] [is a] (capability)[28]
$P^A_{1,20}$: (Self-management capability)[27] [is a] (capability)[28]
$P^A_{1,21}$: (Operational state)[29] [is a] (state)[c,7]
$P^A_{1,22}$: (Use circumstance)[31] [is a] (circumstance)[32]
$P^A_{1,23}$: (Learning curve)[33] [is a] (curve)[34]
$P^A_{1,24}$: (Learning curve)[33] [is proportional with] the (time)[c,8]
$P^A_{1,25}$: (Customer support)[36] [is a] (support)[37]
$P^A_{1,26}$: (Learning time)[38] [is a] (time)[c,8]
$P^A_{1,27}$: (Good training)[39] [is a] (training)[40]
$P^A_{1,28}$: (Heterogeneous user interface)[42] [is a] (user interface)[43]
$P^A_{1,29}$: (Complex programming)[44] [is a] (programming)[c,9]
$P^A_{1,30}$: (Instruction information)[46] [is an] (information)[c,10]
$P^A_{1,31}$: (Different design tasks)[48] [are] (design tasks)[49]
$P^A_{1,32}$: (Complex outcomes)[50] [are] (outcomes)[c,11]
$P^A_{1,33}$: (Difficult interpretation)[52] [is an] (interpretation)[53]
$P^A_{1,34}$: (Software tools)[54] [belong to] (computational tools)[c,12]
$P^A_{1,35}$: (Software packages)[56] [belong to] (computational tools)[c,12]
$P^A_{1,36}$: (Computational tools)[c,12] [are] (tools)[c,13]
$P^A_{1,37}$: (Toolboxes)[58] [belong to] (computational tools)[c,12]
$P^A_{1,38}$: (Unsolved bug)[59] [is a] (bug)[60]
$P^A_{1,39}$: (Relevant data)[61] [are] (data)[c,4]
$P^A_{1,40}$: (User interface)[43] [in an] (interface)[62]
$P^A_{1,41}$: (Data processing)[c,14] [is a] (processing)[c,15]
$P^A_{1,42}$: (Complete software tool)[65] [is a] (software tool)[54]
$P^A_{1,43}$: (High performance)[c,16] [is a] (performance)[c,17]
$P^A_{1,44}$: (Affordable software tool)[68] [is a] (software tool)[54]
$P^A_{1,45}$: (Step by step assistance)[69] [is a] (guided assistance)[70]
$P^A_{1,46}$: (Guided assistance)[70] [is an] (assistance)[71]
$P^A_{1,47}$: (Multifold data visualization)[72] [is a] (data visualization)[c,18]
$P^A_{1,48}$: (Multi-channel data management)[74] [belongs to] (data management)[75]
$P^A_{1,49}$: (Blended datasets)[76] [are] (datasets)[c,19]
$P^A_{1,50}$: (Combined data)[78] [group] (data)[c,4]
$P^A_{1,51}$: (Permanently accessible software tool)[81] [is a] (software tool)[54]
$P^A_{1,52}$: (Customized software tool)[82] [is a] (software tool)[54]
$P^A_{1,53}$: (Intuitive data analytics tool)[83] [belongs to] (data analytics tools)[c,1]
$P^A_{1,54}$: (Smart data analytics tool)[84] [belongs to] (data analytics tools)[c,1]
$P^A_{1,55}$: (Case-based reasoning)[85] [is a] (reasoning)[86]
$P^A_{1,56}$: (Semantic support)[c,66] [is a] (support)[37]
$P^A_{1,57}$: (Data collection practice)[c,21] [is a] (practice)[89]
$P^A_{1,58}$: (Analytical need)[90] [is a] (need)[c,22]
$P^A_{1,59}$: (Incomplete data)[123] [belong to] (data)[c,4]
$P^A_{1,60}$: (Sophisticated data management)[92] [belong to] (data management)[75]
$P^A_{1,61}$: (Data fusion)[94] [is a] (fusion)[96]
$P^A_{1,62}$: (Computational performance)[95] [is a] (performance)[c,17]
$P^A_{1,63}$: (Human interpretation)[98] [is an] (interpretation)[53]

$P^{A}_{1,64}$: (Human interpretation)[98] [is done by] (designers)[4]
$P^{A}_{1,65}$: (Smart semantics)[99] [are] (semantics)[100]
$P^{A}_{1,66}$: (Procedural reasoning)[101] [is a] (reasoning)[86]
$P^{A}_{1,67}$: (MoL data meaning)[102] [is a] (data meaning)[c,23]
$P^{A}_{1,68}$: (PLC data meaning)[110] [is a] (data meaning)[c,23]
$P^{A}_{1,69}$: (Data meaning)[c,23] [is a] (meaning)[112]
$P^{A}_{1,70}$: (AI)[104] [is an] (intelligence)[c,60]
$P^{A}_{1,71}$: (System learning mechanism)[105] [is a] (learning mechanism)[114]
$P^{A}_{1,72}$: (Learning mechanism)[114] [is a] (mechanism)[c,24]
$P^{A}_{1,73}$: (Context information processing)[106] [is] (processing)[c,15]
$P^{A}_{1,74}$: (Situation awareness)[107] [is an] (awareness)[116]
$P^{A}_{1,75}$: (Strategy development)[108] [is a] (development)[c,25]
$P^{A}_{1,76}$: (System adaptation capability)[109] [is a] (capability)[28]
$P^{A}_{1,77}$: (Data analytics solutions)[120] [are] (solutions)[c,27]
$P^{A}_{1,78}$: (Designers)[c,62] [create] (new products)[c,78]
$P^{A}_{1,79}$: (New product)[c,78] [is a] (product)[c,3]
$P^{A}_{1,80}$: (AI)[104] [includes] (reasoning)[86]
$P^{A}_{1,81}$: (AI)[104] [includes] (knowledge representation)[124]
$P^{A}_{1,82}$: (Knowledge representation)[124] [is a] (representation)[125]
$P^{A}_{1,83}$: (AI)[104] [includes] (learning)[c,28]
$P^{A}_{1,84}$: (AI)[104] [includes] (natural language processing)[127]
$P^{A}_{1,85}$: (Natural language processing)[127] [is a] (processing)[c,15]
$P^{A}_{1,86}$: (Machine learning)[c,59] [belongs to] (AI)[104]
$P^{A}_{1,87}$: (Data processing)[c,14] [processes] (data)[c,4]
$P^{A}_{1,88}$: (Product-associated data)[12] [is a] (data)[c,4]
$P^{A}_{1,89}$: (Design task)[c,64] [is a] (task)[c,80]
$P^{A}_{2,1}$: (Big data)[c,29] [is] a (data)[c,4]
$P^{A}_{2,2}$: (Huge data amount)[3] [is a] (data amount)[148]
$P^{A}_{2,3}$: (Various data types)[142] [are] (data types)[c,33]
$P^{A}_{2,4}$: (Hidden pattern)[20] [is] a (pattern)[c,47]
$P^{A}_{2,5}$: (Application context)[22] [is] a (context)[c,67]
$P^{A}_{2,6}$: (Big data analytics)[c,39] [belongs to] (data analytics)[c,48]
$P^{A}_{2,7}$: (Advanced big data analytics)[23] [belongs to] (big data analytics)[c,39]
$P^{A}_{2,8}$: (Intelligent computized solution)[24] [is] a (computized solution)[151]
$P^{A}_{2,9}$: (Computized solution)[151] [is] a (solution)[c,27]
$P^{A}_{2,10}$: (Sentiment analysis)[25] [is] an (analysis)[c,49]
$P^{A}_{2,11}$: (User’s opinion)[28] [is] an (opinion)[154]
$P^{A}_{2,12}$: (Affective state)[30] [is] a (state)[c,7]
$P^{A}_{2,13}$: (Cloud computing service)[32] [is] a (computing service)[156]
$P^{A}_{2,14}$: (Computing service)[156] [is] a (service)[27]
$P^{A}_{2,15}$: (Big data analysis requirement)[33] [is] a (data analysis requirement)[213]
$P^{A}_{2,16}$: (Data analysis requirement)[213] [is] a (requirement)[157]
$P^{A}_{2,17}$: (Online data processing)[36] [is] a (data processing)[c,14]
$P^{A}_{2,18}$: (Diverse data sources)[c,58] [are] (data sources)[c,52]
$P^{A}_{2,19}$: (Diverse data types)[147] [are] (data types)[c,33]
$P^{A}_{2,20}$: (Variable signal inputs)[146] [are] (signal inputs)[37]

$P^{A}_{2,21}$: (Signal input)[37] [is] an (input)[159]
$P^{A}_{2,22}$: (Real life application)[38] [is] an (application)[c,26]
$P^{A}_{2,23}$: (High speed storage)[39] [is] a (storage)[161]
$P^{A}_{2,24}$: (High speed data processing)[40] [is] a (data processing)[c,14]
$P^{A}_{2,25}$: (Online adaptation)[44] [is] an (adaptation)[162]
$P^{A}_{2,26}$: (Contextual element)[45] [is] an (element)[163]
$P^{A}_{2,27}$: (User-specific element)[46] [is] an (element)[163]
$P^{A}_{2,28}$: (Decision-making mechanism)[48] [is] a (mechanism)[c,24]
$P^{A}_{2,29}$: (ML technique)[50] [is] a (computational technique)[c,41]
$P^{A}_{2,30}$: (Computational technique)[c,41] [is] a (technique)[c,50]
$P^{A}_{2,31}$: (ML approach)[c,59] [is] a (computational approach)[167]
$P^{A}_{2,32}$: (Computational approach)[167] [is] an (approach)[166]
$P^{A}_{2,33}$: (Patterns modeling)[52] [consists of] (patterns)[c,47]
$P^{A}_{2,34}$: (Patterns modeling)[52] [is] a (modeling)[c,51]
$P^{A}_{2,35}$: (Correlation modeling)[53] [consists of] (correlations)[169]
$P^{A}_{2,36}$: (Correlation modeling)[53] [is] a (modeling)[c,51]
$P^{A}_{2,37}$: (Unseen data)[55] [is] a (data)[c,4]
$P^{A}_{2,38}$: (Supervised learning)[57] [is] a (learning)[c,28]
$P^{A}_{2,39}$: (Labelled data)[58] [is] a (data)[c,4]
$P^{A}_{2,40}$: (Unsupervised learning)[59] [is] a (learning) [c,28]
$P^{A}_{2,41}$: (Reinforcement learning)[60] [is] a (learning) [c,28]
$P^{A}_{2,42}$: (Goal oriented learning)[61] [is] a (learning) [c,28]
$P^{A}_{2,43}$: (Dynamic situation)[62] [is] a (situation)[171]
$P^{A}_{2,44}$: (Regression technique)[c,42] [is] a (computational technique)[c,41]
$P^{A}_{2,45}$: (Density estimation method)[57] [is] an (estimation method)[173]
$P^{A}_{2,46}$: (Estimation method)[173] [is] a (method)[c,6]
$P^{A}_{2,47}$: (Dimensionality reduction approach)[66] [is] a (reduction approach)[214]
$P^{A}_{2,48}$: (Reduction approach)[214] [is] an (approach)[166]
$P^{A}_{2,49}$: (Human information processing mechanism)[69] [is] a (processing mechanism)[174]
$P^{A}_{2,50}$: (Processing mechanism)[174] [is] a (mechanism)[c,24]
$P^{A}_{2,51}$: (Human information reasoning mechanism)[70] [is] a (reasoning mechanism)[175]
$P^{A}_{2,52}$: (Reasoning mechanism)[175] [is] a (mechanism)[c,24]
$P^{A}_{2,53}$: (CI technique)[71] [is] a (computational technique)[c,41]
$P^{A}_{2,54}$: (Complex data source)[143] [is] a (data source)[c,52]
$P^{A}_{2,55}$: (Uncertain data source)[144] [is] a (data source)[c,52]
$P^{A}_{2,56}$: (Complex real-world data driven problem)[145] [is] a (real-world data driven problem)[72]
$P^{A}_{2,57}$: (Real-world data driven problem)[72] [is] a (data driven problem)[177]
$P^{A}_{2,58}$: (Data driven problem)[177] [is] a (problem)[c,77]
$P^{A}_{2,59}$: (Mathematical modeling)[73] [is] a (modeling)[c,51]
$P^{A}_{2,60}$: (Traditional modeling)[74] [is] a (modeling)[c,51]
$P^{A}_{2,61}$: (High complex process)[75] [is] a (complex process)[215]
$P^{A}_{2,62}$: (Complex process)[215] [is] a (process)[c,53]
$P^{A}_{2,63}$: (Highly uncertain process)[76] [is] an (uncertain process)[216]
$P^{A}_{2,64}$: (Uncertain process)[216] [is] a (process)[c,53]
$P^{A}_{2,65}$: (Highly stochastic process)[77] [is] a (stochastic process)[217]

$P^{A}_{2,66}$: (Stochastic process)[217] [is] a (process)[c,53]
$P^{A}_{2,67}$: (EA)[79] [is] an (algorithm)[c,44]
$P^{A}_{2,68}$: (ANN)[80] [belongs to] (NN)[180]
$P^{A}_{2,69}$: (DNN)[126] [belongs to] (NN)[180]
$P^{A}_{2,70}$: (CNN)[127] [belongs to] (NN)[180]
$P^{A}_{2,71}$: (Imprecise data)[83] [is] a (data)[c,4]
$P^{A}_{2,72}$: (Uncertain data)[84] [is] a (data)[c,4]
$P^{A}_{2,73}$: (Qualitative data)[c,20] [is] a (data)[c,4]
$P^{A}_{2,74}$: (Adaptive control)[86] [is] a (control)[181]
$P^{A}_{2,75}$: (Linguistic qualifier)[87] [is] a (qualifier)[182]
$P^{A}_{2,76}$: (Uncertain real-world)[89] [is] a (real-world)[183]
$P^{A}_{2,77}$: (Uncertain user defined concept)[90] [is] a (user defined concept)[184]
$P^{A}_{2,78}$: (User defined concept)[184] [is] a (concept)[c,54]
$P^{A}_{2,79}$: (Human interpretable fuzzy rule)[91] [is] a (fuzzy rule)[186]
$P^{A}_{2,80}$: (Fuzzy rule)[186] [is] a (rule)[c,76]
$P^{A}_{2,81}$: (Genetic algorithm)[94] [is] an (algorithm)[c,44]
$P^{A}_{2,82}$: (Genetic programming)[95] [is] an (programming)[c,9]
$P^{A}_{2,83}$: (Swarm intelligence optimization algorithm)[96] [is] an (optimization algorithm)[189]
$P^{A}_{2,84}$: (Optimization algorithm)[189] [is] an (algorithm)[c,44]
$P^{A}_{2,85}$: (Complex real-world system)[97] [is] a (real-world system)[191]
$P^{A}_{2,86}$: (Real-world system)[191] [is] a (system)[c,55]
$P^{A}_{2,87}$: (Complex real-world process)[98] [is] a (real-world process)[192]
$P^{A}_{2,88}$: (Real-world process)[192] [is] a (process)[c,53]
$P^{A}_{2,89}$: (Experiential data)[100] [is] a (data)[c,4]
$P^{A}_{2,90}$: (Integrated solution)[103] [is] a (solution)[c,27]
$P^{A}_{2,91}$: (Effective multipurpose intelligent data analysis)[105] [belongs to] (intelligent data analysis)[193]
$P^{A}_{2,92}$: (Intelligent data analysis)[193] [belongs to] (data analysis)[194]
$P^{A}_{2,93}$: (Effective decision-making)[106] [is] a (decision-making)[c,61]
$P^{A}_{2,94}$: (Important feature identification)[108] [belongs to] (feature identification)[195]
$P^{A}_{2,95}$: (Important features)[109] [are identified within] (important feature identification)[108]
$P^{A}_{2,96}$: (Integrated feature)[103] [is] a (feature)[196]
$P^{A}_{2,97}$: (Spatial co-relations identification)[111] [is] a (co-relations identification)[197]
$P^{A}_{2,98}$: (Co-relations identification)[197] [is] a (relation identification)[198]
$P^{A}_{2,99}$: (Relations identification)[198] [is] an (identification)[c,56]
$P^{A}_{2,100}$: (Feature identification)[195] [is] an (identification)[c,56]
$P^{A}_{2,101}$: (Spatial co-relations)[112] [are] (co-relations)[200]
$P^{A}_{2,102}$: (Spatial co-relations)[112] [are identified within] (spatial co-relations identification)[111]
$P^{A}_{2,103}$: (Co-relations)[200] [are] (relations)[c,71]
$P^{A}_{2,104}$: (Temporal co-relations identification)[115] [is] a (co-relations identification)[197]
$P^{A}_{2,105}$: (Temporal co-relations)[116] [are identified within] (temporal co-relations identification)[115]
$P^{A}_{2,106}$: (Temporal co-relations)[116] [are] (co-relation)[200]

$P^{A}_{2,107}$: (Input variable)[113] [is] a (variable)[202]
$P^{A}_{2,108}$: (Given time)[114] [is] a (time)[c,8]
$P^{A}_{2,109}$: (Input parameter)[117] [is] a (parameter)[204]
$P^{A}_{2,110}$: (Overtime)[118] [is] a (time)[c,8]
$P^{A}_{2,111}$: (Large-scale data)[120] [is] a (data)[c,4]
$P^{A}_{2,112}$: (DL approach)[119] [is] a (computational approach)[167]
$P^{A}_{2,113}$: (Spatial correlation)[121] [is] a (correlation)[169]
$P^{A}_{2,114}$: (Temporal correlation)[122] [is] a (correlation)[169]
$P^{A}_{2,115}$: (Multiple hidden layers)[123] [are] (hidden layers)[205]
$P^{A}_{2,116}$: (Hidden layer)[205] [is] a (layer)[206]
$P^{A}_{2,117}$: (Feature learning method)[124] [is] a (learning method)[207]
$P^{A}_{2,118}$: (Learning method)[207] [is] a (method)[c,6]
$P^{A}_{2,119}$: (Supervised approach)[125] [is] an (approach)[166]
$P^{A}_{2,120}$: (Recurrent NN)[128] [belongs to] (NN)[180]
$P^{A}_{2,121}$: (DL)[218] [is] a (learning)[c,28]
$P^{A}_{2,122}$: (DL technique)[129] [is] a (computational technique)[c,41]
$P^{A}_{2,123}$: (Natural language fuzzy rule)[133] [is] a (fuzzy rule)[186]
$P^{A}_{2,124}$: (Hidden relation)[134] [is] a (relation)[c,71]
$P^{A}_{2,125}$: (User-friendly system)[136] [is] a (system)[c,55]
$P^{A}_{2,126}$: (Excellent data analysis tool)[137] [is] a (data analysis tool)[c,1]
$P^{A}_{2,127}$: (High dimensionality)[138] [is] a (dimensionality)[209]
$P^{A}_{2,128}$: (Big data analytics technique)[43] [is] a (data analytics technique)[c,57]
$P^{A}_{2,129}$: (Data analytics technique)[c,57] [is] a (computational technique)[c,41]
$P^{A}_{2,130}$: (Big data analytics method)[41] [is] a (data analytics method)[c,5]
$P^{A}_{2,132}$: (Interpretable knowledge)[42] [is] a (knowledge)[c,2]
$P^{A}_{2,133}$: (CI techniques combination)[101] [combines] (CI techniques)[71]
$P^{A}_{2,134}$: (Offline data processing)[104] [is] a (data processing)[c,14]
$P^{A}_{3,1}$: (Big data technologies)[113] [are used to analyze] (big data)[c,29]
$P^{A}_{3,2}$: (Analytical techniques)[19] [include] (qualitative techniques)[12]
$P^{A}_{3,3}$: (Analytical techniques)[19] [include] (quantitative techniques)[13]
$P^{A}_{3,4}$: (Data organization)[3] [is executed using] (analytical techniques)[19]
$P^{A}_{3,5}$: (Data variety)[8] [increases] (data complexity)[71]
$P^{A}_{3,6}$: (Historic data)[22] [is] a (data type)[c,33]
$P^{A}_{3,7}$: (Scientific data)[79] [is] a (data type)[c,33]
$P^{A}_{3,8}$: (Bibliographic data)[80] [is] a (data type)[c,33]
$P^{A}_{3,9}$: (Performing analytics over data)[39] [requires] (experience)[31]
$P^{A}_{3,10}$: (Data analytics outputs)[51] [are expected from] (data analytics tools)[c,1]
$P^{A}_{3,11}$: (High performance)[c,16] [is expected from] (data analytics tools)[c,1]
$P^{A}_{3,13}$: (Data generation speed)[72] [complicates] (data collection)[c,21]
$P^{A}_{3,14}$: (Data analytics tools)[c,1] [allow] (data mining)[115]
$P^{A}_{3,15}$: (Data mining)[115] [is done using] (algorithmic concepts)[46]
$P^{A}_{3,16}$: (Data mining)[115] [yields] (patterns)[c,47]
$P^{A}_{3,17}$: (Data mining)[115] [is based on] (algorithms)[c,44]
$P^{A}_{3,18}$: (Organization)[c,40] [aim at producing] (value)[c,38]
$P^{A}_{3,19}$: (Data privacy)[109] [is crucial for] (organizations)[c,40]
$P^{A}_{3,20}$: (Data security)[110] [is crucial for] (organizations)[c,40]

$P^{A}_{3,21}$: (Dataset size)[69] [grows based on] (data type)[c,33]
$P^{A}_{3,22}$: (Uncertainty)[95] [influences] (data relevance)[101]
$P^{A}_{3,23}$: (Imprecision)[96] [influences] (data relevance)[101]
$P^{A}_{3,24}$: (Untruth)[99] [fakes] (outputs)[c,46]
$P^{A}_{3,25}$: (Data modeling)[106] [is used for] (data organization)[3]
$P^{A}_{3,26}$: (Data modeling)[106] [is included in] (data pre-processing)[38]
$P^{A}_{3,27}$: (Data organization)[3] [is included in] (data pre-processing)[38]
$P^{A}_{3,28}$: (Data modeling)[106] [creates] (simplified structures)[173]
$P^{A}_{3,29}$: (Simplified structure)[173] [is a] (structure)[78]
$P^{A}_{3,30}$: (Data analytics)[c,48] [can start with] (data modeling)[106]
$P^{A}_{3,33}$: (Traditional data analytics techniques)[119] [belong to] (data analytics techniques)[c,57]
$P^{A}_{3,36}$: (Large dataset)[117] is a (dataset)[c,19]
$P^{A}_{3,37}$: (Raw data)[120] [are] (data)[c,4]
$P^{A}_{3,38}$: (Data behavior)[10] [is a] (behavior)[c,69]
$P^{A}_{3,39}$: (Qualitative technique)[12] [is a] (technique)[c,50]
$P^{A}_{3,40}$: (Quantitative technique)[13] [is a] (technique)[c,50]
$P^{A}_{3,41}$: (Actionable insight)[121] [is an] (insight)[c,45]
$P^{A}_{3,42}$: (Customer preference)[15] [is a] (preference)[124]
$P^{A}_{3,43}$: (Data analytics tools)[c,1] [is a] (computational tool)[c,12]
$P^{A}_{3,45}$: (Analytical technique)[19] [is a] (technique)[c,50]
$P^{A}_{3,46}$: (New insight)[128] [is an] (insight)[c,45]
$P^{A}_{3,47}$: (Probability analysis)[23] [are] (analysis)[c,49]
$P^{A}_{3,48}$: (Classified data association development)[129] [is a] (data association development)[25]
$P^{A}_{3,49}$: (Data association development)[25] [is a] (development)[c,25]
$P^{A}_{3,50}$: (Categorized data association development)[131] [is a] (data association development)[25]
$P^{A}_{3,51}$: (Descriptive analytics)[18] [is an] (analytics][149]
$P^{A}_{3,52}$: (Predictive analytics)[20] [is an] (analytics][149]
$P^{A}_{3,53}$: (Prescriptive analytics)[21] [is an] (analytics][149]
$P^{A}_{3,54}$: (Future outcome)[132] [is an] (outcome)[c,11]
$P^{A}_{3,55}$: (Happenings time frame)[30] [is a] (time frame)[133]
$P^{A}_{3,56}$: (Time frame)[133] is associated with (time)[c,8]
$P^{A}_{3,57}$: (Best possible outcome)[135] [is an] (outcome)[c,11]
$P^{A}_{3,58}$: (Possible outcome)[136] [is an] (outcome)[c,11]
$P^{A}_{3,59}$: (Specific domains)[32] [is a] (domain)[137]
$P^{A}_{3,60}$: (Specific process)[33] [is a] (process)[c,53]
$P^{A}_{3,61}$: (Analytics project)[35] [is a] (project)[139]
$P^{A}_{3,62}$: (Problem identification)[36] [is an] (identification)[c,56]
$P^{A}_{3,63}$: (Data pre-processing)[38] [happens before] (data processing)[c,14]
$P^{A}_{3,66}$: (Data attributes)[43] are (attributes)[143]
$P^{A}_{3,67}$: (Fixed data format)[144] [is a] (data format)[44]
$P^{A}_{3,68}$: (Algorithmic concept)[46] is a (concept)[c,54]
$P^{A}_{3,69}$: (Model-based recommendation)[50] is a (recommendation)[146]
$P^{A}_{3,70}$: (Data analytics output)[51] [is an] (output)[c,46]

$P^{A}_{3,71}$: (Data visualization)[c,18] [is a] (visualization)[147]
$P^{A}_{3,72}$: (Advanced data analytics technique)[148] [is a] (data analytics technique)[c,57]
$P^{A}_{3,73}$: (Behavioral analytics)[52] [is an] (analytics][149]
$P^{A}_{3,74}$: (Social media analytics)[53] [is an] (analytics][149]
$P^{A}_{3,75}$: (Time series analysis)[54] [is an] (analytics][149]
$P^{A}_{3,76}$: (Transactional application)[60] [is an] (application)[c,26]
$P^{A}_{3,77}$: (Real time)[63] is a (time)[c,8]
$P^{A}_{3,78}$: (Large scale)[64] [is a] (scale)[151]
$P^{A}_{3,79}$: (Data generation speed)[72] [is a] (speed)[152]
$P^{A}_{3,80}$: (Data delivery speed)[73] [is a] (speed)[152]
$P^{A}_{3,81}$: (Definite patterns)[153] [is a] (pattern)[c,47]
$P^{A}_{3,82}$: (Relational database management system)[76] [is a] (management system)[154]
$P^{A}_{3,83}$: (Management system)[154] [is a] (system)[c,55]
$P^{A}_{3,84}$: (Pre-defined structure)[156] [is a] (structure)[78]
$P^{A}_{3,85}$: (Different structures)[157] [is a] (structure)[78]
$P^{A}_{3,86}$: (Graph data structures)[81] [is a] (structure)[78]
$P^{A}_{3,87}$: (Standard structure)[158] [is a] (structure)[78]
$P^{A}_{3,88}$: (Operating system level log)[86] [is a] (log)[159]
$P^{A}_{3,89}$: (Data mix)[85] [are] (data)[c,4]
$P^{A}_{3,90}$: (Huge information)[c,70] [is an] (information)[c,10]
$P^{A}_{3,91}$: (Multiple data types)[161] [are] (data types)[c,33]
$P^{A}_{3,92}$: (Multiple data sources)[162] [are] (data sources)[c,52]
$P^{A}_{3,93}$: (Multiple data formats)[163] [are] (data formats)[44]
$P^{A}_{3,94}$: (Missing value)[164] [is a] (value)[c,38]
$P^{A}_{3,95}$: (Missing statement)[165] [is a] (statement)[98]
$P^{A}_{3,96}$: (Huge data challenge)[166] [is a] (data challenge)[91]
$P^{A}_{3,97}$: (High data quality)[167] [is a] (data quality)[7]
$P^{A}_{3,98}$: (Huge dataset)[168] [is a] (dataset)[c,19]
$P^{A}_{3,99}$: (Huge dataset)[168] [is bigger than] (large dataset)[117]
$P^{A}_{3,100}$: (Significant information)[c,75] [is an] (information)[c,10]
$P^{A}_{3,101}$: (Results visualization)[107] [is a] (visualization)[147]
$P^{A}_{3,102}$: (Data collection)[c,21] [is related to] (data)[c,4]
$P^{A}_{3,103}$: (Data alignment)[104] [is related to] (data)[c,4]
$P^{A}_{3,104}$: (Data transformation)[105] [is related to] (data)[c,4]
$P^{A}_{3,105}$: (Data modeling)[106] [is related to] (data)[c,4]
$P^{A}_{3,106}$: (Data modeling)[106] [is a] (modeling)[c,51]
$P^{A}_{3,107}$: (Data privacy)[109] [is related to] (data)[c,4]
$P^{A}_{3,108}$: (Data quality)[7] [is related to] (data)[c,4]
$P^{A}_{3,109}$: (Data governance)[111] [is related to] (data)[c,4]
$P^{A}_{3,110}$: (Data security)[110] [is related to] (data)[c,4]
$P^{A}_{3,111}$: (Traditional data processing systems)[170] [is a] (data processing system)[112]
$P^{A}_{3,112}$: (Data processing system)[112] [is a] (system)[c,55]
$P^{A}_{3,113}$: (Big data technologies)[113] [is a] (technology)[c,74]
$P^{A}_{4,1}$: (Design problem)[1] [is a] (problem)[c,77]
$P^{A}_{4,2}$: (Product information)[53] [is an] (information)[c,10]
$P^{A}_{4,3}$: (Product knowledge)[54] [is a] (knowledge)[c,2]

$P^A_{4,4}$: (Purposive novelty)[55] [is a] (novelty)[13]
$P^A_{4,5}$: (A difficult design problem)[56] [is a] (design problem)[1]
$P^A_{4,6}$: (Obvious solution)[57] [is a] (solution)[c,27]
$P^A_{4,7}$: (Creative solution)[58] [is a] (solution)[c,27]
$P^A_{4,8}$: (Knowledge-based system)[59] [is a] (system)[c,55]
$P^A_{4,9}$: (High-level decision-making)[60] [is a] (decision-making) [c,61]
$P^A_{4,10}$: (Creative decision-making)[61] [is a] (decision-making) [c,61]
$P^A_{4,12}$: (Decision-makers)[c,62] [miss] (product information)[53]
$P^A_{4,13}$: (Decision-making) [c,61] [solves a] (design problem)[1]
$P^A_{4,14}$: (Knowledge)[c,2] [is crucial for] (creativity)[12]
$P^A_{4,15}$: (Creativity)[12] [is to be considered to solve] (design problems)[1]
$P^A_{4,16}$: (Creativity)[12] [impacts] (design process)[19]
$P^A_{4,17}$: (Design process)[19] [exists to solve] (design problems)[1]
$P^A_{4,18}$: (Design process)[19] [is a] (process)[c,53]
$P^A_{4,19}$: (Decision-making process)[64] [is a] (process)[c,53]
$P^A_{4,20}$: (Semantic interpretation)[c,66] [is an] (interpretation)[c,72]
$P^A_{4,21}$: (Robust decision-making)[67] [is a] (decision-making)[c,61]
$P^A_{4,22}$: (Key influencing event)[29] [is an] (event)[c,79]
$P^A_{4,23}$: (Appropriate inquiries)[30] [is an] (inquiry)[69]
$P^A_{4,24}$: (Context-driven decision-making)[70] [is a] (decision-making)[c,61]
$P^A_{4,25}$: (Dynamically integrated knowledge)[71] [is a] (knowledge)[c,2]
$P^A_{4,26}$: (Up-to-date data sources)[36] [are] (data sources)[c,52]
$P^A_{4,27}$: (Proactive decision-making)[42] [is a] (decision-making)[c,61]
$P^A_{4,28}$: (Unmanned systems)[44] [is a] (system)[c,55]
$P^A_{4,29}$: (Smart reasoning technology)[45] [is a] (reasoning technology)[47]
$P^A_{4,30}$: (Reasoning technology)[47] [is a] (technology)[c,74]
$P^A_{4,31}$: (Real-time data)[73] [is a] (data)[c,4]
$P^A_{4,32}$: (Autonomous agents)[49] [are] (agents)[75]
$P^A_{4,33}$: (Decision-support system)[76] [is a] (system)[c,55]
$P^A_{4,34}$: (New context)[77] [is a] (context)[c,67]
$P^A_{4,35}$: (Actionable decisions)[78] [are] (decisions)[51]
$P^A_{4,36}$: (Decision-making)[c,61] [happens in a] (context)[c,67].
$P^A_{5,1}$: (Mental activity)[23] [is] (an activity)[14]
$P^A_{5,2}$: (Physical activity)[53] [is] (an activity)[14]
$P^A_{5,3}$: (Human factor)[54] [is a] (factor)[24]
$P^A_{5,4}$: (Machine factor)[55] [is a] (factor)[24]
$P^A_{5,5}$: (Interaction factor)[56] [is a] (factor)[24]
$P^A_{5,6}$: (Human behaviour)[57] [is a] (behaviour)[c,69]
$P^A_{5,7}$: (Smart system)[58] [is a] (system)[c,55]
$P^A_{5,8}$: (Adaptive systems)[59] [is a] (system)[c,55]
$P^A_{5,9}$: (Decentralized systems)[60][is a] (system)[c,55]
$P^A_{5,10}$: (Distributed systems)[61] [is a] (system)[c,55]
$P^A_{5,11}$: (Multi-scale systems)[62] [is a] (system)[c,55]
$P^A_{5,12}$: (Varied components)[63] [is a] (component)[30]
$P^A_{5,13}$: (Internal relations)[64] [is a] (relation)[c,71]
$P^A_{5,14}$: (External relations)[65] [is a] (relation)[c,71]

$P^{A}_{5,15}$: (Physical connectivity)[66] [is a] (connectivity)[32]
$P^{A}_{5,16}$: (Syntactic interaction)[67] [is an] (interaction)[18]
$P^{A}_{5,18}$: (Human mental process)[69] [is a] (process)[c,53]
$P^{A}_{5,19}$: (Semantic technology)[70] [is a] (technology)[c,74]
$P^{A}_{5,20}$: (Computational actor)[71] [is an] (actor)[37]
$P^{A}_{5,21}$: (CPSs interactions)[72] [is an] (interaction)[18]
$P^{A}_{5,22}$: (Traditional HCI)[73] [belong to] (HCI)[9]
$P^{A}_{5,23}$: (Traditional HMI)[74] [belong to] (HMI)[8]
$P^{A}_{5,24}$: (Internal interrelationships)[75] [are] (interrelationships)[40]
$P^{A}_{5,25}$: (External interrelationships)[76] [are] (interrelationships)[40]
$P^{A}_{5,26}$: (Diverse interactions)[77] [are] (interactions)[18]
$P^{A}_{5,27}$: (Intelligent system)[78] [is a] (system)[c,55]
$P^{A}_{5,28}$: (Intelligence-based SHI)[79] [belongs to] (SHI)[12]
$P^{A}_{5,29}$: (Human role)[10] [is a] (role)[c,81]

Appendix 5

Representation of normal and faulty behavior of middle-of-life data streams implemented in Matlab

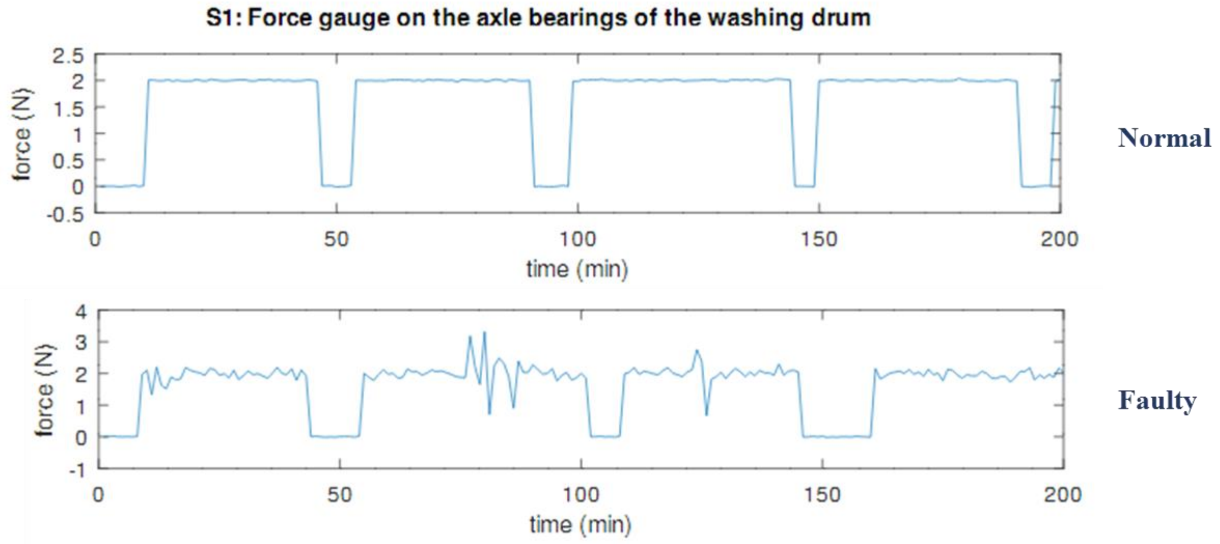


Figure 1: Representation of normal and faulty behavior of sensor number 1

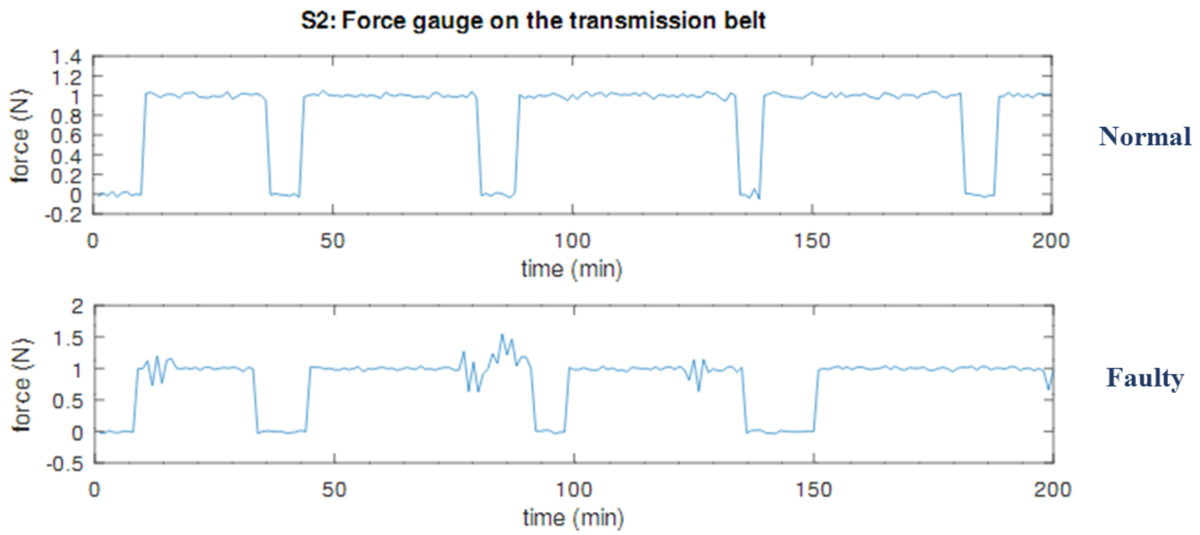


Figure 2: Representation of normal and faulty behavior of sensor number 2

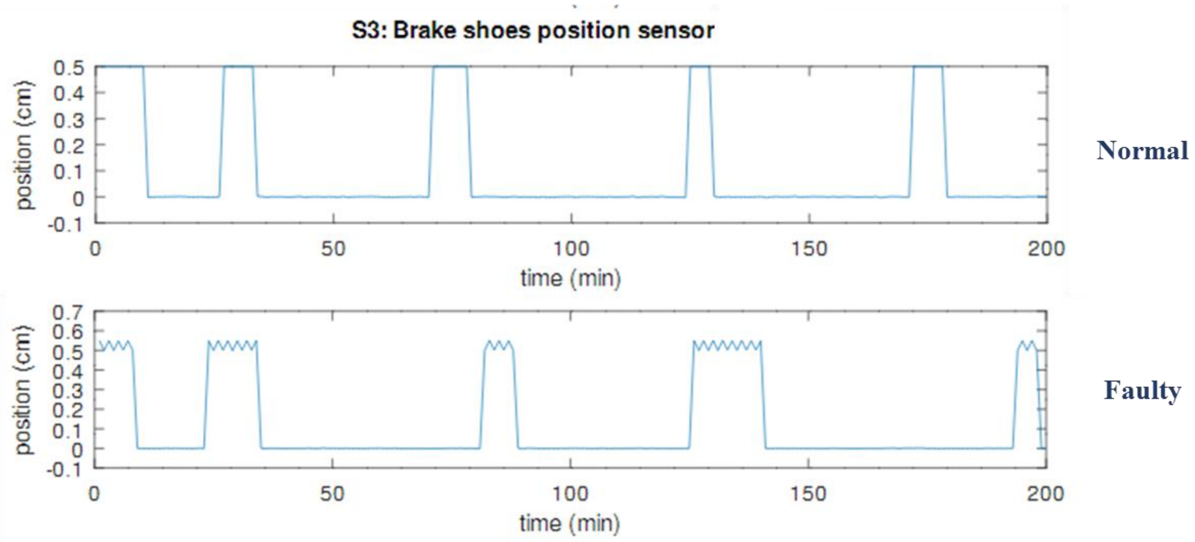


Figure 3: Representation of normal and faulty behavior of sensor number 3

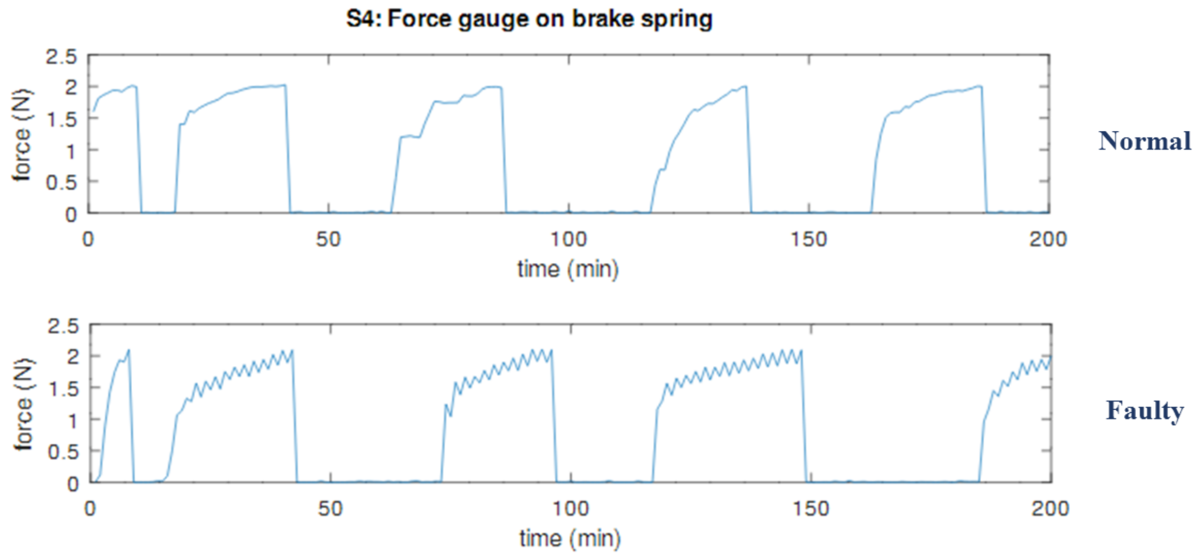


Figure 4: Representation of normal and faulty behavior of sensor number 4

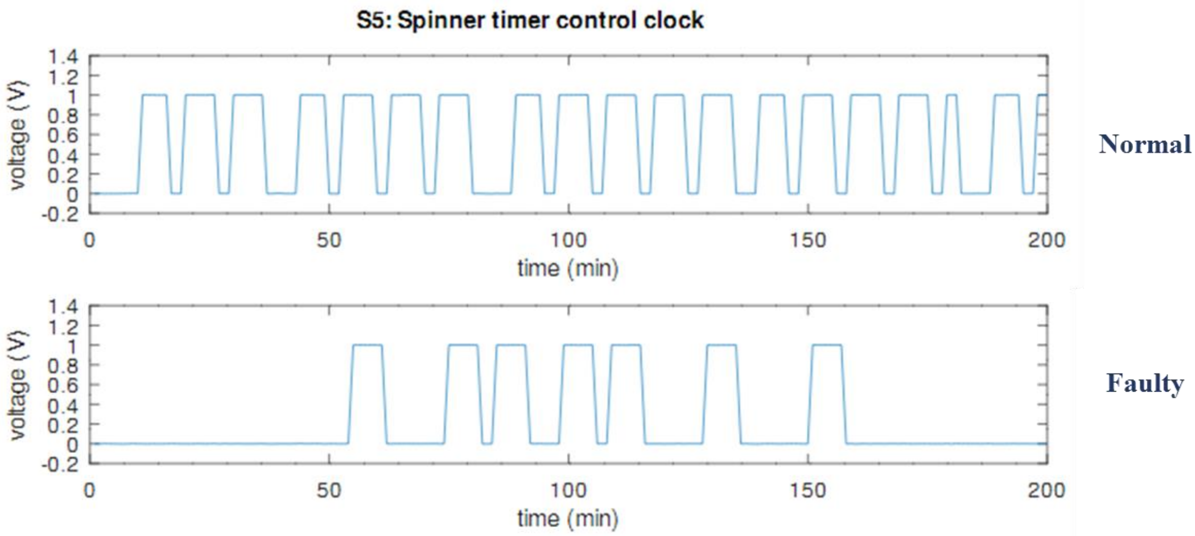


Figure 5: Representation of normal and faulty behavior of sensor number 5

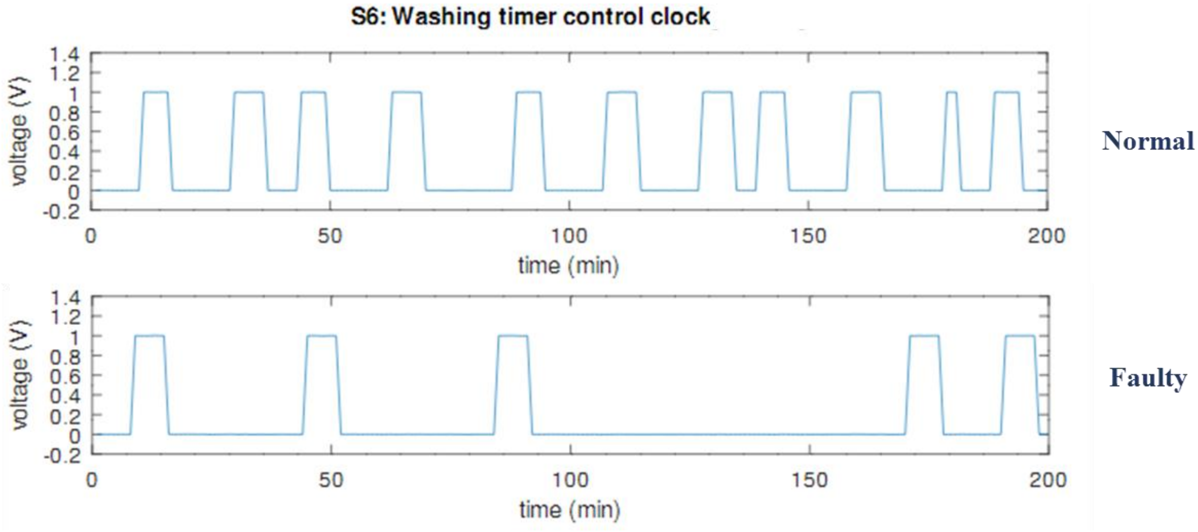


Figure 6: Representation of normal and faulty behavior of sensor number 6

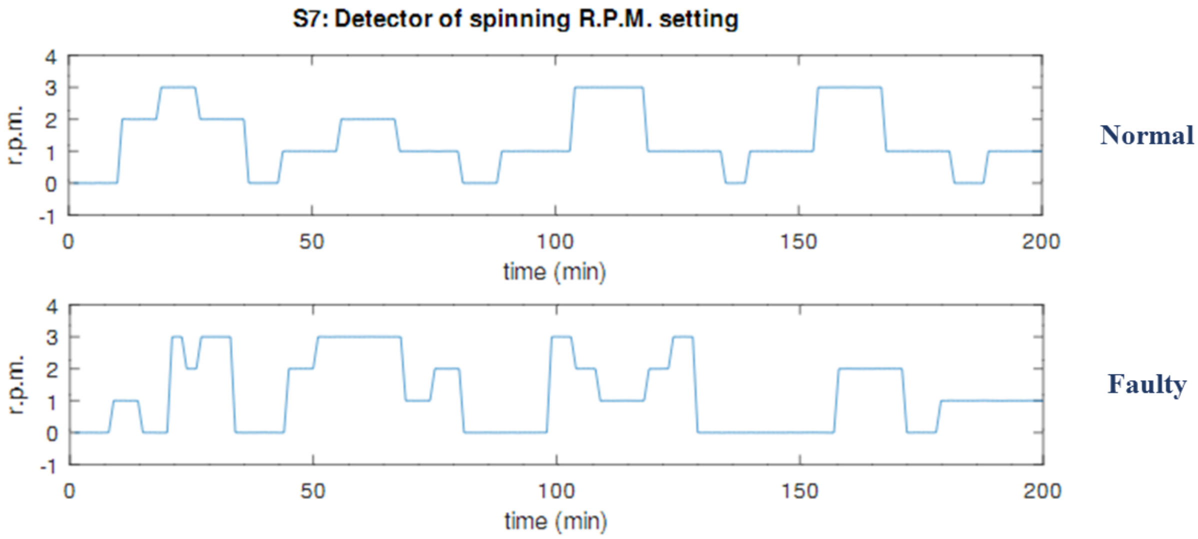
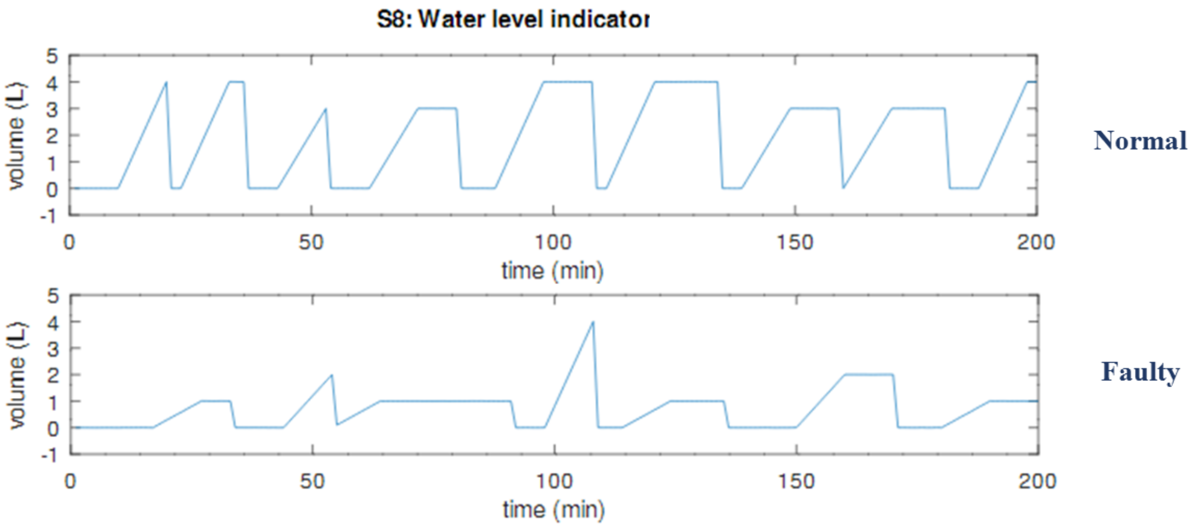
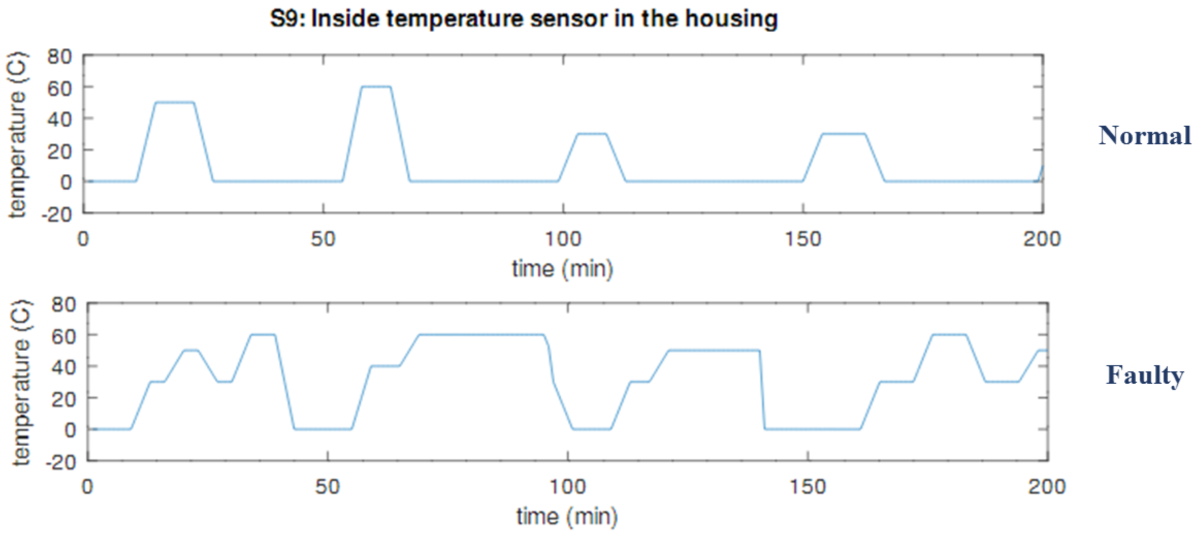


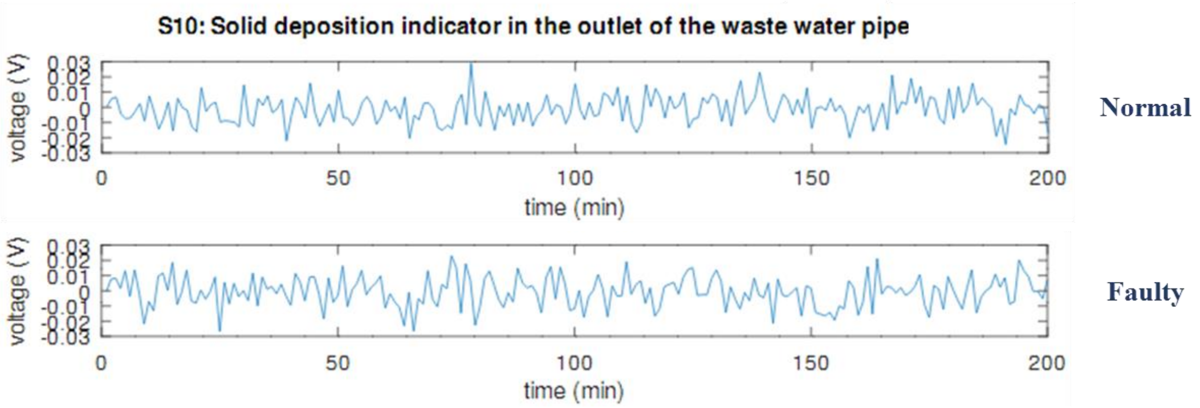
Figure 7: Representation of normal and faulty behavior of sensor number 7



<u>Figure 8: Representation of normal and faulty behavior of sensor number 8</u>



<u>Figure 9: Representation of normal and faulty behavior of sensor number 9</u>



<u>Figure 10:</u> Representation of normal and faulty behavior of sensor number 10

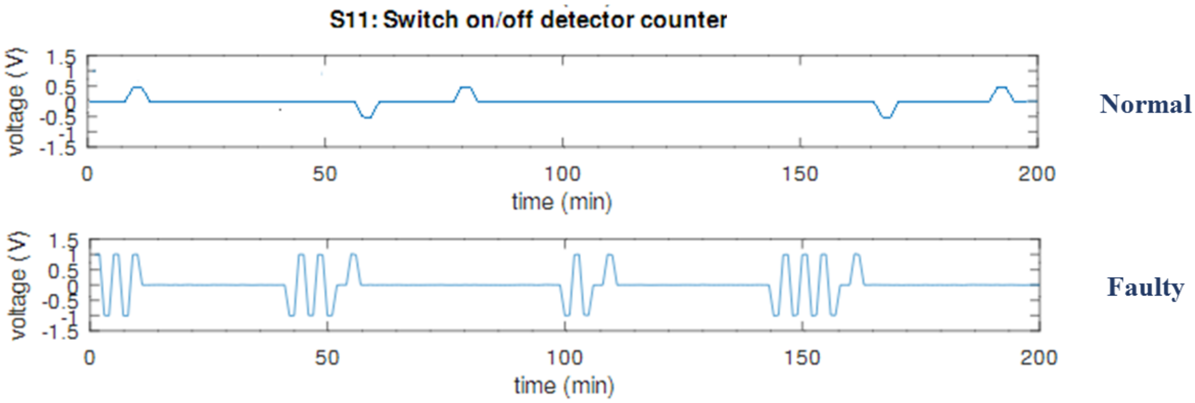


Figure 11: Representation of normal and faulty behavior of sensor number 11

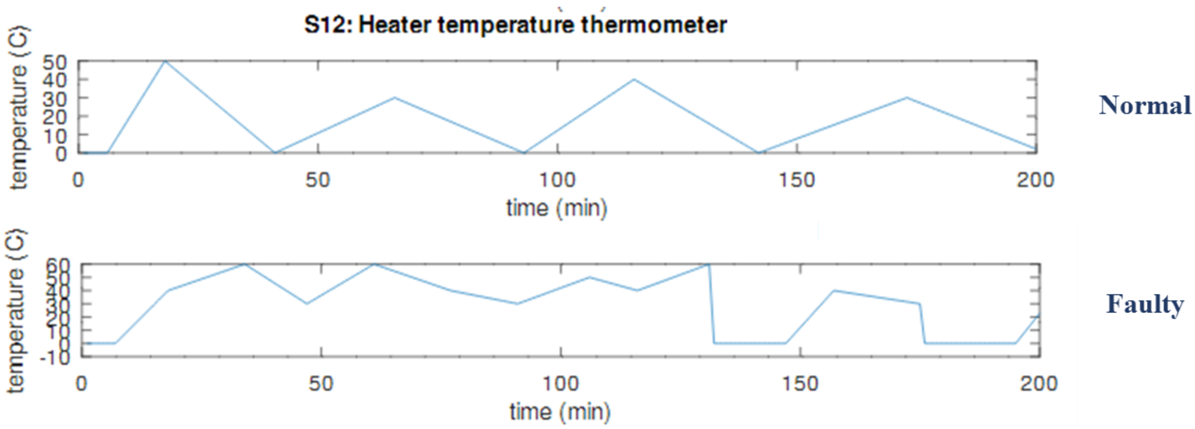


Figure 12: Representation of normal and faulty behavior of sensor number 12

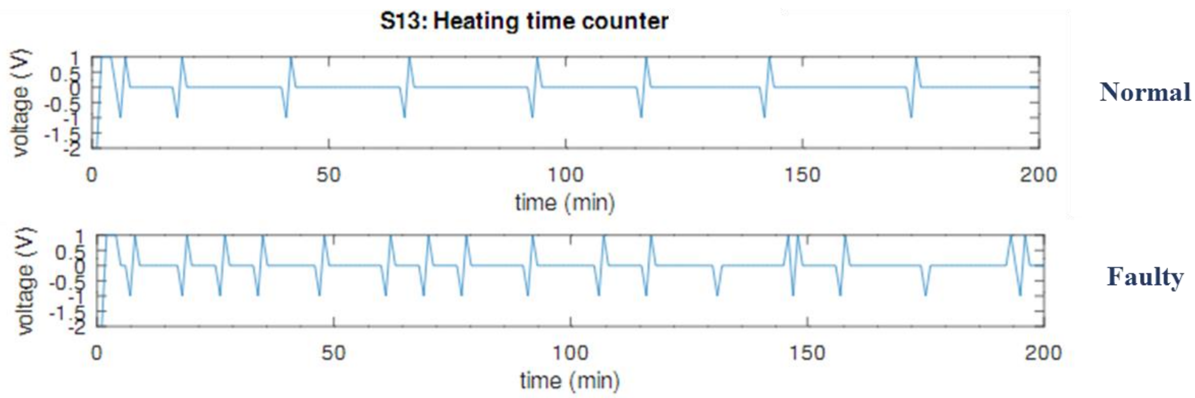


Figure 13: Representation of normal and faulty behavior of sensor number 13

Biography

Fatima-Zahra Abou Eddahab is born in Meknes, Morocco on the 20 of August 1990. In 2009, she received her Bachelor degree in Mathematics and Physics from Omer Ibn Abdelaziz Institute, Morocco. In 2013, she obtained her Master degree in Mechanical Engineering – Design and Integrated Production from Mohammadia School of Engineers, Morocco. In the same year, she moved to France to follow a second Master in Industrial Engineering – Product Development that she obtained at the end of 2014 from Grenoble Institute of Technology. She was recruited for a year until end 2015 by the same institute to model agile decision-making in innovation processes and study the contribution of mixed performances in such processes. In 2016 she moved to the Netherlands, where she did her Ph.D. project for four years at the Faculty of Industrial Design Engineering at Delft University of Technology. This project was part of an EU project called Falcon H2020. Her current research interests include data analytics; smart data processing tools; smart products augmentation; products development.