Data analytics pipeline for prediction and decision making in complex products and systems development

Opiyo, Eliab

Publication date
2015

Document Version
Final published version

Published in
Proceedings of the ASME 2015 International Mechanical Engineering Congress & Exposition IMECE2015

Citation (APA)
Opiyo, E. (2015). Data analytics pipeline for prediction and decision making in complex products and systems development. In *Proceedings of the ASME 2015 International Mechanical Engineering Congress & Exposition IMECE2015* (Vol. 11, pp. 1-11). ASME.

Proceedings of the ASME 2015 International Mechanical Engineering Congress & Exposition
IMECE2015
November 13-19, 2015, Houston, Texas, USA

IMECE2015-53296

DATA ANALYTICS PIPELINE FOR PREDICTION AND DECISION MAKING IN COMPLEX PRODUCTS AND SYSTEMS DEVELOPMENT

Eliab Z. Opiyo
Faculty of Industrial Design Engineering
Delft University of Technology
Landbergstraat 15, NL-2628 CE Delft
The Netherlands
Email: e.z.opiyo@io.tudelft.nl

Keywords: Data analytics, forecasting, complex products and systems, design space exploration.

ABSTRACT

Facilitating data analytics for effective prediction in complex products or systems development is the focus of the research described in this paper. The specific objective was to develop strategies and a data analytics pipeline with a view to supporting exploration of the design space of complex products or systems upfront. The underlying challenges tackled included how to acquire and store raw data gathered by using both the traditional methods and advanced Internet of Things (IoT) devices, how to preprocess and transform raw data into a form suited for data analytics, and how to deal with analytics. A pipeline for data analytics to support decision making in complex products or systems development is proposed and its applicability illustrated with a practical example. The incorporation of advanced analytics techniques into the proposed pipeline allows users to acquire data and to insightfully and intelligently predict aspects such as cost and assembly time early on, and to make decisions based on data that may otherwise deemed to be inaccessible or unusable. This work contributes to the efforts directed toward applying data analytics techniques in a way that can have a profound impact on an engineering product or system development process.

1. INTRODUCTION

There is a huge excitement around the promise of using advanced data analytics techniques against large and diverse datasets in various application domains. Engineering product development is one of the application domains which advanced data analytics practices can positively impact. Upfront prediction of aspects such as performance, reliability, and cost is one of the early phase engineering design analysis activities in which advanced data analytics techniques such as machine learning and data mining could play role in, e.g., to facilitate enriching, converting, and capturing trends in data (i.e., data acquired using traditional data gathering techniques coupled with e.g., advanced data acquisition tools such as Internet of Things (IoT) devices) to gain insights. Products or systems[1] (with mechanical, electrical, software, cyber and/or sensing elements in them) are increasingly becoming more and more complex. To a large extent, this can be attributed to continuous advances of various advanced technologies used in them, including computing, communication, sensing, and control technologies—which are increasingly becoming intertwined with products or systems. In general terms, the more complex a product or system becomes, the harder the prediction of the aspects such as performance, reliability, or cost becomes. The motivation behind the work reported in this paper was the general understanding that the success of a product or system during its use phase will depend on how well the developers can predict and understand the dynamics of the system upfront, and be able to accommodate the knowledge gained into the development process and to decisively address any uncovered problems early on.

This work builds and expands on our earlier work—see, e.g., [1], [2], [3], in which a roadmap for prediction of future prospects and for exploration of design space of novel complex systems such as cyber-physical systems (CPS) was created—see Figure 1. This roadmap outlines a systematic approach for identification of the components and features of complex products or systems such as CPSs, and for modeling of these products or systems with a view to forecasting aspects such as

[1] The term 'product' is used in this paper to refer to an object or service created as a result of an engineering process and serves a need or satisfies a set of requirements while the term 'system' is used to refer a collection of objects or things working together as parts of a mechanism.

cost, reliability, performance, and assembly time early on. It specifies the four steps of complex product or system design space exploration process, which are: (a) needs analysis, (b) components and features identification, (c) modeling and representation, and (d) exploration (i.e., engineering analyses and data analytics). In the latter step, among other things—apart from carrying out various engineering analyses such as Finite Element Analysis (FEA) or Computational Fluid Dynamics (CFD) analysis to optimize complex designs, the data needed in making predictions is gathered and suitable methods are chosen and applied to prognosticate future prospects. To this end, several data analytics techniques may be applied together to achieve reliable prediction from massive design datasets, which can in turn be used as the basis for optimizing complex designs in real time. In this work, we focused specifically on the fourth step of this roadmap (i.e., the exploration—i.e., engineering analyses and data analytics step) and we attempted to develop a pipeline and strategies for data analytics with a view to supporting prediction based on massive product datasets and decision making processes in the early stages of the engineering product or system development processes. Thus, the principal objective of the reported research was to develop techniques for acquiring data in raw form, for preprocessing and transforming gathered raw data into a form suited for data analytics, and for dealing with analytics—with a view to predicting aspects such as assembly time, performance, reliability, and cost. The intention here was to ultimately come up with an all-inclusive pipeline that combines several tools and applications to manage the entire analytics process, i.e., from data acquisition through to analytics. The challenges we attempted to address included how to acquire data in raw form, how to preprocess and transform gathered raw data into a form suited for data analytics, and how to deal with analytics. The sought after strategies and pipeline were intended to provide a comprehensive strategic and operational plan for data-processing, as well as tools to effectively support the product or system developers to deal with the above-mentioned design space exploration and data analytics challenges.

In this paper, we describe the research we conducted and present the results. The paper is organized as follows. We first present a concise literature review in the following Section. Then, we describe the needs and present the requirements for the data analytics pipeline, present the proposed data analytics pipeline, and illustrate how the pipeline and the proposed strategies can be put to use. We finally discuss the research results and present some conclusions about the findings of the research.

2. A CONCISE LITERATURE REVIEW

Data analytics is increasingly becoming an important subject in a number of research and application domains. Today, analytics tools are used variously in many application domains to transform data into useful insights, including, e.g., in business—where analytics methods and tools are used to process data and embed information into business processes [4], in weather forecasting [5], and in engineering [6].

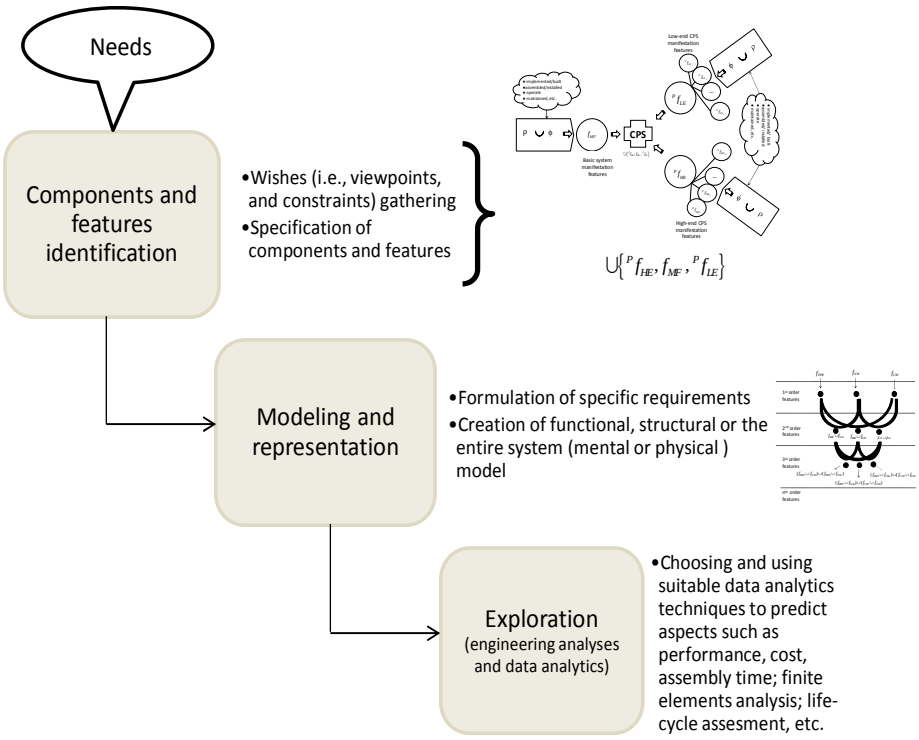

Figure 1 Upfront design space exploration scheme, see also [1].

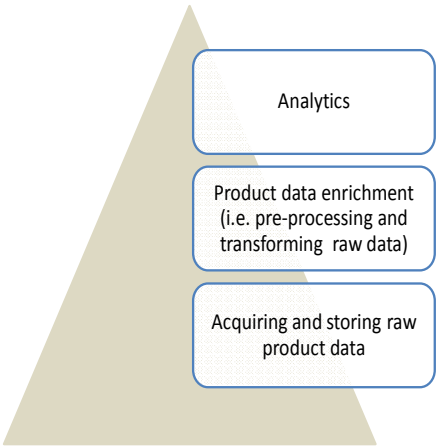

Figure 2 A data analytics workflow showing a chain of data-processing stages.

Numerous methods and techniques from various disciplines are used to deal with the challenges faced in data analytics, e.g., in acquiring data and in analyzing previously untapped data sources to gain new insights and to uncover knowledge [4], [7]. These include visual analytics [6], text analytics [8], predictive analytics [9], descriptive analytics [10], diagnostic analytics [11], and prescriptive analytics [12]. Furthermore, methods and techniques such as machine learning [13], data mining [14], [15], statistical analyses [15], and natural language processing [16] are also applied to process, analyze, and to discover patterns in data. Data analytics methods and techniques such as those mentioned above allow us to make good use of data datasets originating from various sources, including, for instance, from scientific experiments, computer-based services, online sales, or from machine measurements in shop floors, services, or any other activities and to obtain insight into complex operations.

The data dealt with in analytics include *text* data (i.e., text analytics)—see, e.g., [17] in which approaches such as statistical pattern learning are used to derive information from patterns and trends in texts, *video* data (i.e., video analytics), see, e.g., [18]—where video analytics applications are used to monitor scenes to gain insights and to understand the actions occurring, and *graphical* data (i.e., graph analytics)—see, e.g. [19] which enables us to explore 'many-to-many' complex relationships and to map relationships among high volumes of highly connected data to find connections, to unlock insightful questions, and to produce more accurate outcomes.

Overall, regardless of the domain in which analytics is applied or of the form or type of data, the analytics processes typically pass through three major stages, namely: *acquiring* and storing raw data—much of which is generated in large

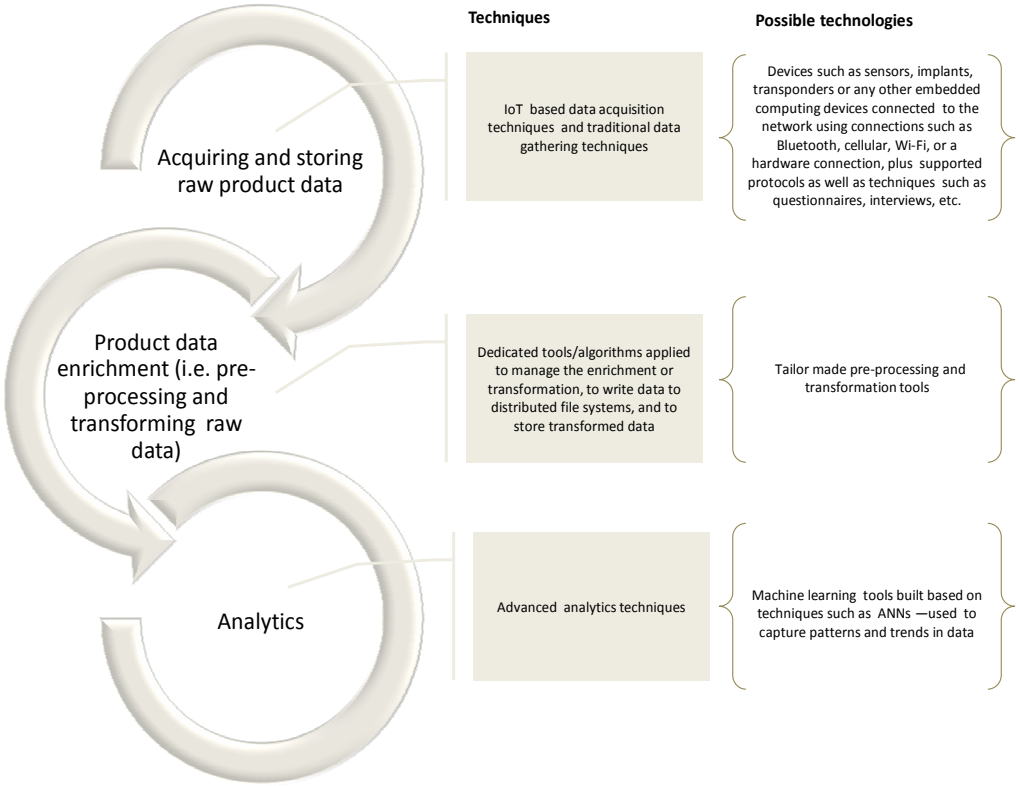

Figure 3 Techniques and possible technologies to support data analytics in complex systems development processes.

scale and in real time, typically coming from sensors, embedded computing devices, cameras (i.e., video and photos), audio recording devices, web-based processes, social media applications, log files, office applications, or from traditional transactional applications—see, e.g., [20]; *preprocessing and transformation* of raw gathered data, and *analyzing* gathered data (Figure 2). Various approaches are used to collect data in raw form and to manipulate the collected data (i.e., to inspect, clean, and to transform data)—see Figure 3. The most commonly used approaches and means to acquire data are: *registration*—i.e., depositing data or information into a specified repository, *questionnaires*—i.e., forms are filled in by respondents; this method is typically used to collect regular or infrequent routine data, which can be adopted for the entire population or sampled population, *interviews*—i.e., information is obtained through inquiry and recorded by the enumerators; and this is performed either by using survey forms (i.e., structured interviews), or by taking notes while speaking with respondents (i.e., during open interviews), *observations*—i.e., taking direct measurements during practical laboratory experiments, or by using *IoT devices* such as sensors or any embedded computing devices—e.g., to measure a physical phenomenon, or properties such as force, temperature, or light intensity.

In light of the above review, it can be said that there are already numerous techniques in place that can be adopted and applied in the context of the research reported in this paper to acquire data in raw form, to preprocess and transform gathered raw data into a form suited for data analytics, and to deal with analytics. The choice of a specific technique to use will depend on the circumstances on the ground, including e.g., the type of data dealt with.

3. NEEDS ANALYSIS AND REQUIREMENTS FOR A DATA ANALYTICS PIPELINE

A typical data analysis assignment involves gathering, manipulating (i.e., inspecting, cleaning, transforming), visualizing, and modeling data with a view to capturing and fostering an understanding of the patterns and trends in data. The requirements for an analytics system can be identified by considering the underlying data handling sub-processes and tasks involved. A suite of data analytics support tools must:

- Facilitate acquisition and storage of raw data.
- Support ubiquitous data acquisition by using IoT devices such as sensors and embedded computing devices within the existing Internet infrastructure, protocols, and a wide range of open source platforms.
- Support processing of the raw data acquired by using the traditional data gathering techniques as well as the data acquired by using advanced data acquisition tools such as sensors, embedded computing devices, and other IoT devices.
- Support transformation of raw data in the event data is missing, requiring enrichment, or if there is a need to transform the way the values are represented.
- Facilitate data sharing among various stakeholders within the existing Internet infrastructure.
- Enable fast, effective, and comprehensive data analysis by using advanced data analytics techniques.
- Facilitate the discovery of useful information from raw datasets and visualization of processed data, suggest conclusions, and support decision-making.

Apart from these high-level functional requirements, according to ISO/IEC 9126 quality standard—see e.g., [21], [39], the data analytics software components must also attain the desirable efficiency, maintainability, portability, usability and other desirable quality characteristics. A suite of software tools, consisting of a mix of dedicated tools and some selected everyday applications can be stringed together and used to meet the above described data analytics requirements.

4. PROPOSED DATA ANALYTICS PIPELINE

One of the principal functional requirements for the pipeline is to support users to perform analytics on the acquired raw data with a view to extracting useful insights. Building a pipeline for carrying out scalable analytics on voluminous and high velocity datasets acquired by using IoT devices or through traditional data gathering techniques coming in in many varied formats is one of the challenges that need to be addressed. The pipeline in its final form will combine several tailor made and existing applications to support activities in the following broad data analytics steps: (1) acquiring and storing of raw product data, (2) preprocessing and transformation of data, and (3) data analytics (Figure 4). The two pre-exploration steps shown in Figure 1, namely, the *components and features identification* step—in which the needs, wishes, and the preliminary general requirements for the product or system are identified and used as the basis for identifying the components and features, and the *modeling and representation* step—in which concrete and more specific requirements are formulated and used as the basis for modeling or representing the product or system variously, precede these three broad analytics steps.

Acquiring and storing raw product data is the initial step of the proposed data analytics workflow. To this end, the traditional data gathering techniques such as observation, interview, and questionnaire survey can be adopted and used. Also, myriad protocols can facilitate raw data acquisition by using IoT devices (such as sensors, implants, transponders, or any other embedded computing devices). These devices may connect to the network using connections such as Bluetooth, cellular, Wi-Fi, or a hardware connection, which would send messages using defined protocols. Protocols widely supported by IoT applications such as Message Queue Telemetry Transport (MQTT)[2][22], Constrained Application Protocol (CoAP) [23], Extensible Messaging and Presence Protocol (XMPP) [24], and others can be applied. Considerations in choosing a protocol to use include ubiquity and the ability to provide a wide range of support. Using any chosen protocol

[2] Mosquitto—see, e.g., [29], is one of the widely used open source MQTT brokers which could be used.

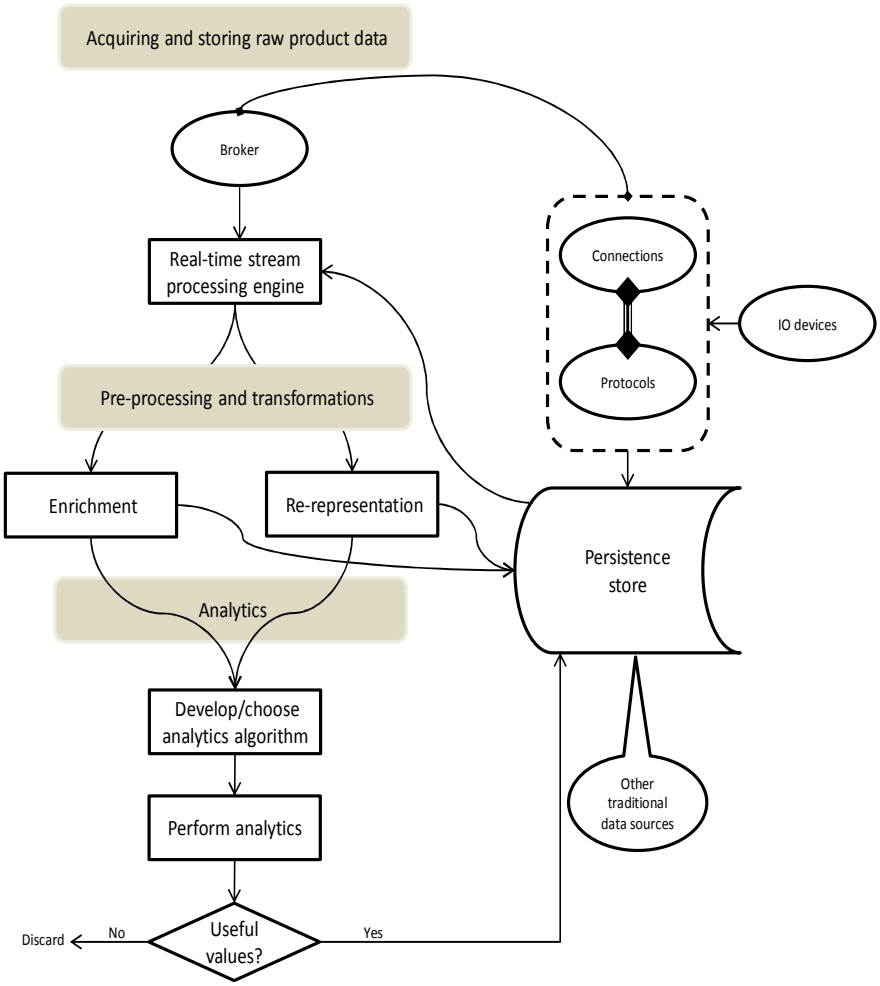

Figure 4 Broad data analytics steps.

will help to get the messages in hand—i.e. to catch and represent events or observations from connected devices in the event data is acquired by using connected IoT devices. Once a message is received by a broker, it can be handed over to an analytics system for the downstream activities of preprocessing and data transformation. It is recommended that the original source data must be stored before any preprocessing or performing transformation. This can be very helpful down the road, e.g., should there be any debugging issues during transformation or preprocessing, or if there is a need to replay a sequence of messages, e.g., for analysis or testing purposes.

There are several options for storing data acquired by using IoT devices. These include using data warehousing infrastructures or frameworks such as Hadoop [25] and Hive [26], or working with NoSQL document databases [27] such as Couchbase [28]. The key storage requirements include the ability to offer a good combination of high-throughput and low-latency characteristics, schema-less feature, the ability to handle high data volume input, flexibility, easiness to add new data, the ability to store very large datasets reliably and to stream those datasets at high bandwidth to clients, and the ability to write data directly to a distributed file system.

Custom code to enable continuous writing and storage of data can be written and appended to the message broker. Codes can also be written to enable pushing messages to an intermediate messaging brokers or moving messages to different elements of the pipeline. One possible strategy is to have in place separate mechanisms for writing raw data to a persistence store and for moving the data into a real-time stream processing engine.

As far as *preprocessing and transformation* step is concerned, it is important to recognize that raw product data derived from IoT devices or gathered by using traditional methods may not necessarily be suited for analytics. In certain circumstances, data may be missing—thus requiring enrichment, or representation of values may need to be altered. Consequently, there is always the need for a preprocessing and transformation step to manage enrichment and re-representation

of gathered raw data. However, it is imperative to have in place strategies and mechanisms for storing both the enriched and re-represented product datasets as well as for storing the original raw product dataset. It should be noted that preprocessing and transformation can be very expensive processes. They may also add significant latency to the data analytics workflow. From the implementation perspective, preprocessing, transformation, and data storage can be institutionalized in several different ways. Frameworks or systems such as Pig [30] and Storm [31] can be used in batch mode analysis and in writing data to a distributed file system. In this way, both the original raw data as well as the transformed or enriched data can be stored for future use. One of the key requirements in preprocessing and transformation is to achieve low-latency characteristics. In this regard, it is important to note that running multiple jobs in sequence will obviously add a lot of latency to the workflow.

Data analytics is the final step of the pipeline, which starts once the data has been preprocessed or transformed. The goal here is to analyze data (i.e., large amounts of data - typically obtained in design office environments) with a view to capturing trends and patterns in data. An appropriate advanced analytics tool needs to be selected or developed and used to handle data, including continuous streams of data. Such a technique should be able to manage high-volume data streams and capable to perform operations such as event correlation, rolling metric calculations, and aggregate statistical analyses if needed. Examples of advanced analytics techniques that can be used to capture patterns and trends in data include machine learning techniques such as Artificial Neural Networks (ANNs) [32]. Another key requirement is that an analytics tool, as a key element of an analytics pipeline, should be a good fit for working with all sorts of data, including data streamed by IoT devices. Some of the obtained analytics values may sometimes be discarded directly while other values may need to be stored in a persistent store—and it always make sense to keep more data than the data that is discarded. Most of the available tools such as the Waikato Environment for Knowledge Analysis (WEKA) [33] and MATLAB [34], [35] are open source in nature, and this allow users to implement additional analytics algorithms as required—i.e., their functionalities can be accessed from other programming environments, e.g., for WEKA, there is an interface for Python and for the statistical programming language R.

5. ILLUSTRATION

In this section we explain how the proposed data analytics pipeline and strategies can be used in predicting various aspects of an engineering product or system such as cost, reliability, performance, and assembly time. Consider the product development assignment summarized in Figure 5.

The hand-operated noodle making machine (Figure 6) needs to be conceptualized (i.e., its components and features identified) and then modelled or appropriately represented first. Once the product model or representation (i.e., the hand-operated noodle making machine model or representation)—created by taking into consideration usability, performance and

COMPANY'S BACKGROUND

AAER & Co. is a company that develops various kinds of home utensils. It has received an assignment to develop a noodle press machine (see Figure 6). This small machine will be sold in department stores and used mainly domestically. The user should be able either to fix the machine—e.g., on the kitchen table, or to hold it with one hand while operating it. The noodle press machine will be produced in large quantities and should be fairly inexpensive.

ASSIGNMENT

Design the noodle press machine and show analytically (using a suitable parametric mathematical model) that the designed components and connections can withstand the loading when the machine is in use. When designing, take into consideration the need for minimum usage of material (for all components) and make sure that the design meets functional and non- functional requirements for the noodle press machine. The components can take any shapes or appearance. The main dimensions for the noodle press machine are shown in Figure 5 (these dimensions can be changed/varied slightly). Choose suitable materials for the components. Consider possible real scenarios of extreme loading.

Figure 5 An example product development assignment.

other quality requirements—is in place, then aspects such as cost, performance, reliability, or assembly time can be predicted. The feature-based upfront design space exploration scheme—see Figure 1and refer also to [1], [2], [3]—is used to guide the conceptualization, modeling and representation, and exploration processes. Various aspects of the hand-operated noodle making machine can be predicted according to the data analytics workflow we propose in this paper (see Figure 4 and refer also to Figure 3).

According to the feature-based upfront design space exploration scheme (see Figure 1), components and feature

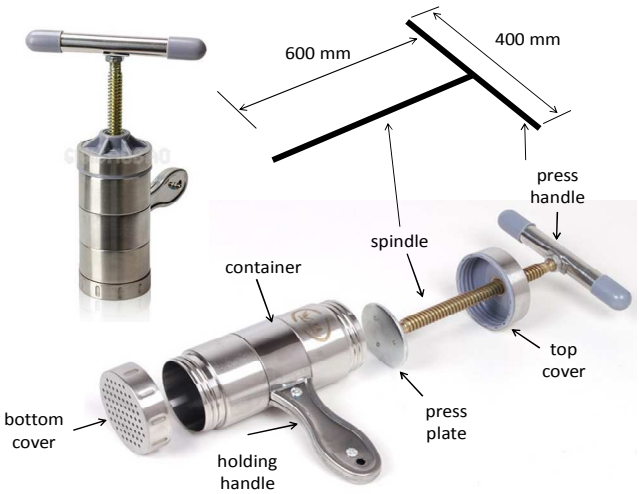

Figure 6 A case-study product (a hand-operated noodle making machine).

identification (also referred to as conceptualization in this article) is the first process step, which is guided by the sub-scheme depicted in Figure 7. This sub-scheme discerns and defines three sets of features that a product or complex system may encompass, which are: (1) set of paradigmatic low-end complexity manifestation (CM) features, ($^{P}f_{LE}$); (2) set of paradigmatic high-end CM features, ($^{P}f_{HE}$); and (3) set of basic-system manifestation features, (f_{MF}). Based on the viewpoints and constraints—gathered through user-centered techniques such as interviews and focus group research, the basic-system as well as CM physical, software, and cyber components can first be identified (see Table 1—but it is important to note here that for the case study product, i.e., the intended hand-operated noodle making machine, there were no any software or cyber components). The process entails identification of functional-physical, software, or cyber components that would make the hand-operated noodle making machine meet its functional as well as other quality requirements, and brainstorming on how the components will operate together and which features will make it operate as desired.

As shown in Table 1, the hand-operated noodle making machine consists of some basic-system manifestation functional components that will enable it to operate and offer services to meet functional and other quality requirements. These basic-system components, among other things, enable the hand-operated noodle making machine to function as required. There were no any specific requirements that called for incorporation of paradigmatic CM functional components in this hand operated machine. To achieve the required functionality, the components with certain particular features must be aggregated in different ways as desired in order to come up with combinations or assemblies of components (i.e., complex mechanisms) that work collectively, and whose combined actions enable the eventual product or system to function as desired. Features (i.e., the elements of f_{MF}, $^{P}f_{LE}$, *and* $^{P}f_{HE}$) that must be incorporated into the components in order for them to function properly must be identified by taking into consideration functional and other quality requirements. The elements of these sets of features may be physical, software, or cyber in nature. In short, it is imperative to recognize that a feature or a given set of features can be manifested on components, and this will allow the hand-operated noodle making machine to function or operate as required. And depending on the types of features incorporated into or onto the components of the hand-operated noodle making machine, it may function or operate differently.

Practically, the process of identifying the features that should be incorporated onto or into the identified components of the hand-operated noodle making machine simply entails specifying the elements of f_{MF}, $^{P}f_{LE}$, and $^{P}f_{HE}$. The applicable paradigmatic features can be identified in several ways, including, for instance, through expert judgement, focus group discussions, or by making associations, analogies, or references to prior similar products or past experiences. The identified features can further be classified differently, e.g., into the sets of form features, functional features, interface features, and so on. Through the above-described process, the functional components and features of a hand-operated noodle making machine summarized in Table 1 were identified.

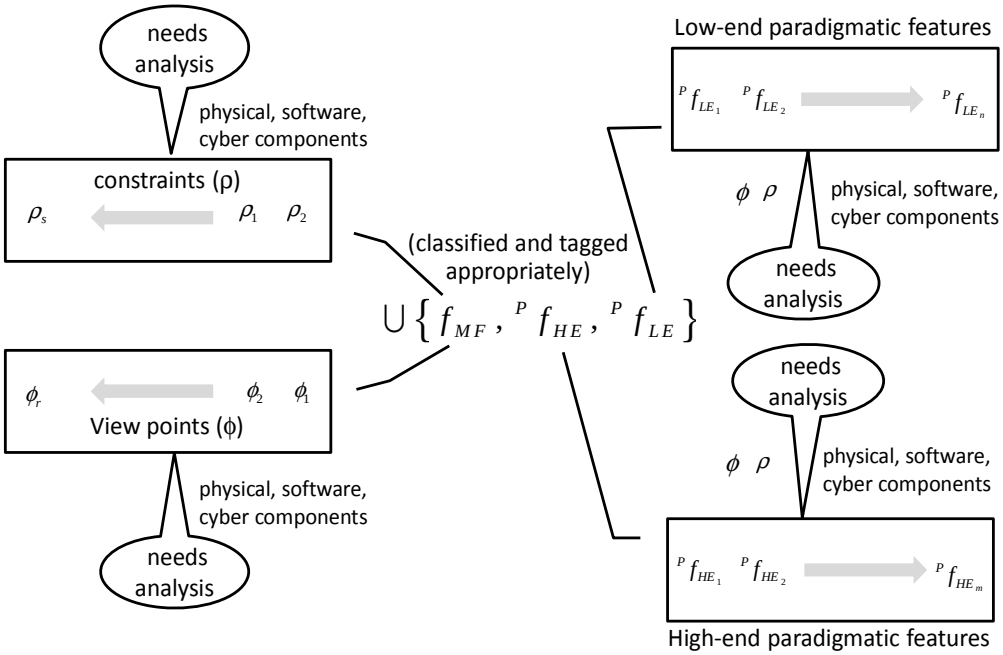

Figure 7 Components and features identification scheme, see also [3].

Table 1 Deriving components and features of the noodle making machine.

	Components (refer also to Figure 5)	Basic-system manifestation features, f_{MF} $f_{MF}=\{f_{MF_1}, f_{MF_2},..., f_{MF_o}\}$		Low-end paradigmatic CM features, ${}^{P}f_{LE}$ ${}^{P}f_{LE}=\{{}^{P}f_{LE_1}, {}^{P}f_{LE_{i2}},..., {}^{P}f_{LE_n}\}$	High-end paradigmatic CM features, ${}^{P}f_{HE}$ ${}^{P}f_{HE}=\{{}^{P}f_{HE_1}, {}^{P}f_{HE_2},..., {}^{P}f_{HE_m}\}$
		Designation	Description		
Hand-operated noodle making machine	Container	f_{MF_1}	Fastening threads (bottom cover interface)	${}^{P}f_{LE}=\varnothing$ (None)	${}^{P}f_{HE}=\varnothing$ (None)
		f_{MF_2}	Fastening thread (top cover interface)		
		f_{MF_3}	Groove (holding handle interface)		
	Spindle	f_{MF_4}	Power thread		
		f_{MF_5}	Fastening thread (press handle interface)		
		f_{MF_6}	Fastening thread (press plate interface)		
	Press handle	f_{MF_7}	Fastening thread (spindle interface)		
	Top cover	f_{MF_8}	Fastening thread (container interface)		
	Press plate	f_{MF_9}	Fastening thread (spindle interface)		
	Holding handle	$f_{MF_{10}}$	Groove (container's handle interface)		
		$f_{MF_{11}}$	Outer hole		
		$f_{MF_{12}}$	Inner hole		
		$f_{MF_{13}}$	Varied cross-section area (gripping)		
	Bottom cover	$f_{MF_{14}}$	Fastening thread (container interface)		
		$f_{MF_{15}}$	Noodle discharging holes		

Based on the sets of the identified components and basic paradigmatic features, a hand-operated noodle making machine can then be modelled or represented according to the scheme shown in Figure 8. Modeling techniques such as Petri Net [36], or low-fidelity prototyping techniques [37] such as abstract prototyping [38], [39], [40], and paper prototyping [41] may be used to represent the product or system architecturally or to model the operations of the product or system. Once the product or system has been modelled, various aspects of the hand-operated noodle making machine such as costs and assembly time can subsequently be predicted. To this end, advanced analytics techniques such as machine learning as well as statistical analysis routines incorporated into the pipeline can then be applied to enable the designers to analyze various datasets to gain insights into different aspects of the hand-operated noodle making machine.

The proposed roadmap (see Figure 1) allows us to conceptualize, to model or represent the product or system early on, and to explore, e.g., in this case, how the hand-operated noodle making machine will be like and how it will function or operate. The resulting models or representations also allow the designers to investigate information, materials, and/or energy flows within the hand-operated noodle making machine and to evaluate the performance, operation, reliability, or any other quality characteristics. In this way, the potential problems can be uncovered and addressed early on. Also, an architectural representation (e.g., a skeleton representation) of the hand-operated noodle making machine can be built and used to visualize the layout of the components. Such a representation can be used as the basis, e.g., for determining the optimal arrangement of components, evaluating the kinematics and dynamics of the system, or as the basis for estimating the cost of the hand-operated noodle making machine—to this end, the appropriate data needed to predict cost, e.g., historical cost data of comparable prior products, should be available in a

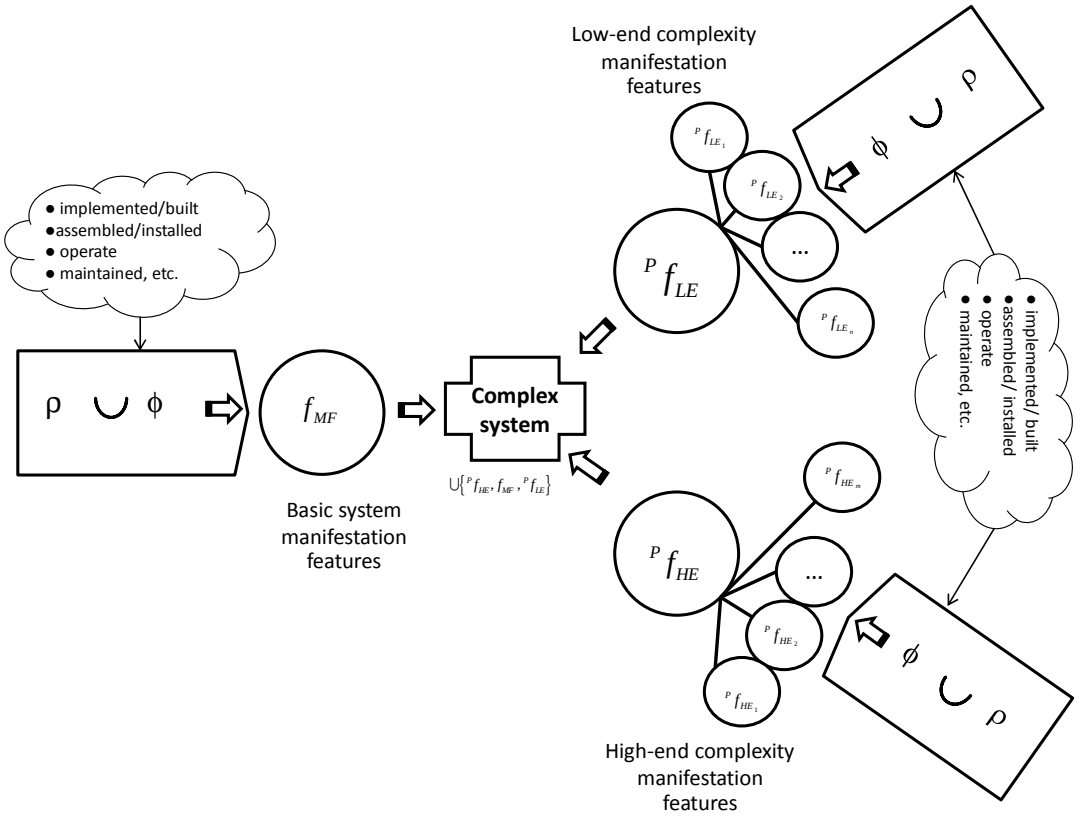

Figure 8 Components and features aggregation scheme, see also [1],[3].

usable form, and suitable prediction tool must be created or selected, and applied.

Overall, specification of the components of the hand-operated noodle making machine in advance—based on knowledge of the design constraints as well as of the requirements for the product or system and wishes of various stakeholders—and identification of the features that the components should have in order for them to operate and to function as desired, are the key pre-exploration sub steps that the design process must go through, particularly when doing analytics in new product development processes. Knowledge of the design constraints and of the requirements for the product or system, as well as of the components and features in them allows the designers to conceptualize, to model or represent the product or system appropriately, and to subsequently use the analytics pipeline proposed in this paper to predict various aspects of the product or system such cost, performance, and assembly time early on (i.e., according to the chain of data-processing steps specified in the pipeline). Such prediction can be effected by comparing the design of the product or system with the designs of prior products or systems—this prediction relies on the availability of prior historical datasets (e.g., based on prior comparable product's or system's cost, performance, or assembly time datasets).

6. SUMMARY, CONCLUSIONS, AND FUTURE WORK

The paper has addressed the issues pertaining to facilitating and supporting data analytics in complex products or systems (with mechanical, electrical, software, cyber, and/or sensing elements) development processes. The challenges we attempted to address included how to acquire data by using both the traditional data acquisition methods and IoT devices, how to preprocess and transform gathered raw data into a form suited for data analytics, and how to deal with analytics. The paper has introduced strategies and a pipeline which consists of a chain of data-processing stages in which advanced analytics techniques are applied. These strategies and the pipeline can be used against large and diverse datasets to predict various aspects of complex product or system. The data analytics pipeline can be scaled or adapted, and used in various application domains to support processing of different types of data such as structured, unstructured, streamed, or batch data of different sizes—including large datasets (i.e., big data[3]) that

[3] According to the literature, a dataset is taken to be 'big data' if it cannot be handled by a traditional relational database management systems (RDMS). For instance, Naveenkumar and Selvavinayagam [20] define big data as 'datasets whose size or type is beyond the ability of traditional relational databases to capture, manage, and process the data with low-latency'. The defining characteristics for big datasets are high-volume, high-velocity, and/or high-variety.

require capabilities beyond those of the traditional relational database management systems (RDBMS).

A case-study application example has been used to illustrate how the pipeline can be used in a practical engineering product or system development process. In short, the proposed pipeline is a constituent of a broader design space exploration workflow scheme designed for prediction of various aspects of complex products or systems upfront. This design space exploration scheme defines a four-tier procedural workflow, which starts with needs analysis, followed by the identification of the components and features of the product or system. The product or system can then be modeled or represented as desired (i.e., the requirements for the product or system need to be formulated first, and then used as the basis for modeling—i.e., building a functional, structural, or a complete product or system model as desired). Modeling in this sense entails aggregating the identified features of the product or system to achieve the required architectural composition and functionality. Modeling techniques such as Petri Net, or low-fidelity prototyping techniques such as abstract prototyping and paper prototyping may be used to model or to represent the architecture or the operations of the intended product or complex system. Various aspects of the product or system such as cost, assembly time, or performance can subsequently be predicted.

The proposed pipeline and strategies allow users to acquire data and to insightfully and intelligently explore the design space of products or systems. The incorporation of powerful advanced analytics techniques such as machine learning as well as various statistical analyses into the pipeline aims to allow the designers to analyze untapped data sources or data that may otherwise deemed to be inaccessible or unusable to gain new insights, resulting in faster exploration of design space and sensible decisions. The reported research contributes to the complex products and systems conceptualization knowledge, in particular to the research efforts on applying data analytics techniques upfront in a way that could have a profound impact on the processes of development of complex engineering products or systems. Apart from complex engineering product or system development, the proposed data analytics workflow can be adapted and used for prediction in other application domains with high societal impact such as health care and business. Future works include delving deeper into issues concerning the validity of the proposed strategies and of the data analytics workflow, including their applicability and manifestation in different practical settings. This would entail developing additional dedicated algorithms that may be needed, e.g., to process or manage high-volume data streams, or to perform certain specific operations such as event correlation, metric calculations, or any other statistical computations needed to make predictions.

REFERENCES

[1]. Opiyo, E. Z. & Horváth, I. (2014), “Feature-Based Prognosis of Performance and Cost Implications of Cyber-Physical Systems: An Illustration of Theory and Process”, In Proc. of the ASME Design Engineering Technical Conferences, Buffalo, NY, USA, August 17-20, 2014, Paper No. DETC2014-34343 (American Society of Mechanical Engineers, New York City, NY, USA).

[2]. Opiyo, E. Z., & Horváth, I. (2014), “Reference scheme for deriving aspects and criteria for forecasting functional performance and cost implications of cyber-physical products”, In Proceedings of the 10th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2014), May 19-23, 2014, Budapest, Hungary, pp. 199 - 214.

[3]. Opiyo, E. Z. A new feature-based approach to upfront complex systems modeling, analysis and prognosis (Submitted, under review), Int. J. Information Technology and Management.

[4]. LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2013). Big data, analytics and the path from insights to value. MIT Sloan Management Review, 21.

[5]. Venayagamoorthy, G., Rohrig, K., & Erlich, I. (2012). One step ahead: short-term wind power forecasting and intelligent predictive control based on data analytics. Power and Energy Magazine, IEEE, 10(5), 70-78.

[6]. Keim, D., Andrienko, G., Fekete, J. D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges (pp. 154-175). Springer Berlin Heidelberg.

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

[8]. Aggarwal, C. C., & Zhai, C. (2012). Mining text data. Springer Science & Business Media

[9]. Maciejewski, R., Hafen, R., Rudolph, S., Larew, S. G., Mitchell, M. A., Cleveland, W. S., & Ebert, D. S. (2011). Forecasting hotspots—A predictive analytics approach. Visualization and Computer Graphics, IEEE Transactions on, 17(4), 440-453.

[10]. Abt, K. (1987). Descriptive data analysis: a concept between confirmatory and exploratory data analysis. Methods Inf Med, 26, 77-88.

[11]. Banerjee, A., Bandyopadhyay, T., & Acharya, P. (2013). Data Analytics: Hyped Up Aspirations or True Potential?. Vikalpa, 30(4), 1-11.

[12]. Basu, A. T. A. N. U. (2013). Five pillars of prescriptive analytics success. Analytics Magazine, 8-12.

[13]. Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 12). New York: springer.

[14]. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37.

[15]. Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer, 27(2), 83-85.

[16]. Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.
[17]. Cunningham, H., Maynard, D., & Bontcheva, K. (2011). Text processing with gate. Gateway Press CA.
[18]. Dore, A., Soto, M., & Regazzoni, C. S. (2010). Bayesian tracking for video analytics. Signal Processing Magazine, IEEE, 27(5), 46-55.
[19]. Xin, R. S., Crankshaw, D., Dave, A., Gonzalez, J. E., Franklin, M. J., & Stoica, I. (2014). GraphX: Unifying Data-Parallel and Graph-Parallel Analytics. arXiv preprint arXiv:1402.2394.
[20]. Naveenkumar, N., & Selvavinayagam, G. (2014). A Survey on Appliance and Secure In Big Data. International Journal for Scientific Research and Development, 1(6), 237-243
[21]. Jung, H. W., Kim, S. G., & Chung, C. S. (2004). Measuring software product quality: A survey of ISO/IEC 9126. IEEE software, 21(5), 88-92.
[22]. Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January). MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on (pp. 791-798). IEEE.
[23]. Shelby, Z., Hartke, K., & Bormann, C. (2014). The Constrained Application Protocol (CoAP).
[24]. Laukkanen, M. (2004). Extensible Messaging and Presence Protocol (XMPP). University of Helsinki Department of Computer Science.
[25]. White, T. (2009). Hadoop: the definitive guide: the definitive guide. “O’Reilly Media, Inc.".
[26]. Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Anthony, S., & Murthy, R. (2009). Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment, 2(2), 1626-1629.
[27]. Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. International Journal of Web Information Systems, 9(1), 69-82.
[28]. Brown, M. C. (2012). Getting Started with Couchbase Server. "O'Reilly Media, Inc.".
[29]. Neisse, R., Steri, G., & Baldini, G. (2014, October). Enforcement of security policy rules for the internet of things. In Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on (pp. 165-172). IEEE.
[30]. Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008, June). Pig latin: a not-so-foreign language for data processing. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1099-1110). ACM.
[31]. Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. ACM SIGKDD Explorations Newsletter, 14(2), 1-5.
[32]. Haykin, S. S., (2009). Neural networks and learning machines (Vol. 3). Upper Saddle River: Pearson Education.
[33]. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.
[34]. Demuth, H., & Beale, M. (1993). Neural network toolbox for use with MATLAB.
[35]. Hanselman, D., & Littlefield, B. C. (1997). Mastering MATLAB 5: A comprehensive tutorial and reference. Prentice Hall PTR.
[36]. Murata, T. (1989). Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77(4), 541-580.
[37]. Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. interactions, 3(1), 76-85.
[38]. Opiyo, E. Z., Horváth, I., & Vergeest, J. S. (2009). Extending the Scope of Quality Assurance of CAD Systems: Putting Underlying Engineering Principles, Theories, and Methods on the Spotlight. Journal of Computing and Information Science in Engineering, 9(2), 024502.
[39]. Opiyo, E. Z. (2003). Facilitating the development of design support software by abstract prototyping. Ph.D. Thesis, TU Delft, Delft University of Technology.
[40]. Opiyo, E. Z., Horváth, I., & Vergeest, J. S. (2000). Software Tools for Abstract Prototyping of Design Support Tools. Proceedings of the 2000 ASME DETC/CIE September, 10-13.
[41]. Sefelin, R., Tscheligi, M., & Giller, V. (2003, April). Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping. In CHI'03 extended abstracts on Human factors in computing systems (pp. 778-779). ACM.