

A Computational Framework for Procedural Abduction Done by Smart Cyber-Physical Systems

Horvath, Imre

DOI

[10.3390/designs3010001](https://doi.org/10.3390/designs3010001)

Publication date

2019

Document Version

Final published version

Published in

Designs

Citation (APA)

Horvath, I. (2019). A Computational Framework for Procedural Abduction Done by Smart Cyber-Physical Systems. *Designs*, 3(1), 1-27. Article 1. <https://doi.org/10.3390/designs3010001>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Article

A Computational Framework for Procedural Abduction Done by Smart Cyber-Physical Systems

Imre Horváth 

Cyber-Physical Systems Design Section, Department of Design Engineering, Faculty of Industrial Design Engineering, Delft University of Technology, Landbergstraat 15, 2628 CD Delft, The Netherlands; i.horvath@tudelft.nl; Tel.: +31-15-278-3520

Received: 11 October 2018; Accepted: 19 December 2018; Published: 25 December 2018



Abstract: To be able to provide appropriate services in social and human application contexts, smart cyber-physical systems (S-CPSs) need ampliative reasoning and decision-making (ARDM) mechanisms. As one option, procedural abduction (PA) is suggested for self-managing S-CPSs. PA is a knowledge-based computation and learning mechanism. The objective of this article is to provide a comprehensive description of the computational framework proposed for PA. Towards this end, first the essence of smart cyber-physical systems is discussed. Then, the main recent research results related to computational abduction and ampliative reasoning are discussed. PA facilitates beliefs-driven contemplation of the momentary performance of S-CPSs, including a ‘best option’-based setting of the servicing objective and realization of any demanded adaptation. The computational framework of PA includes eight clusters of computational activities: (i) run-time extraction of signals and data by sensing, (ii) recognition of events, (iii) inferring about existing situations, (iv) building awareness of the state and circumstances of operation, (v) devising alternative performance enhancement strategies, (vi) deciding on the best system adaptation, (vii) devising and scheduling the implied interventions, and (viii) actuating effectors and controls. Several cognitive algorithms and computational actions are used to implement PA in a compositional manner. PA necessitates not only a synergic interoperation of the algorithms, but also an objective-dependent fusion of the pre-programmed and the run time acquired chunks of knowledge. A fully fledged implementation of PA is underway, which will make verification and validation possible in the context of various smart CPSs.

Keywords: smart cyber-physical systems; self-generated intelligence; ampliative reasoning mechanism; procedural abduction; data-driven system control; run-time acquired data; computational functions; self-adaptation capability; human/socially-centered applications

1. Introduction

1.1. An Evolutionary View on Cyber-Physical Systems

The aspiration for having cyber-physical systems (CPSs) emerged more than 60 years ago [1]. However, the principles of their practical manifestation were formulated just a decade ago [2] and the enabling technologies are becoming available nowadays [3]. At the time of inception, CPSs were seen as large-scale, distributed, self-controlling, software integrated application systems, which combine the functionalities and technological enablers of embedded control systems, advanced robotic systems, networked distributed systems, real-time control systems, and collaborative software agents systems [4]. Their operations were supposed to be coordinated, controlled, and monitored by a digital computing, communication and control core [5]. Interestingly, this novel system paradigm has been going through changes from both theoretical and practical points of view in the last years [6]. And this process will go on [7]. Some envisage even a kind of metamorphosis of the paradigm due to new

technological affordances provided by, for instance, massive data networking technologies, cloud computing, artificial system intelligence, and cognitive system engineering technologies [8–10].

The almost infinite number of networked sources of sensor data, the databases residing on servers in the cloud, the growing availability of massive data flows, the development of smart data analytics approaches, the emergence of run time adapting system controls, and the other technologies shown in Figure 1, are the main enablers of the radical paradigm shift that is already observable in the academic research [11,12]. This shift encourages us to look at smart CPSs as the transformative systems of the near future. Nevertheless, it is worth pointing out that the traditional definitions and interpretations of CPSs are not able to capture the essence of the above circumscribed changes [13]. The main reason is that they place the emphasis on having a predefined and deterministic tight coupling between the physical world and the cyber-world, rather than on achieving synergism between run-time acquired data and dynamic operational objectives [14]. By doing so, they actually restrict the paradigmatic evolution of this family of engineered systems. This is a vital issue since S-CPSs can offer novel functional affordances and services that cannot be provided by other systems [15].

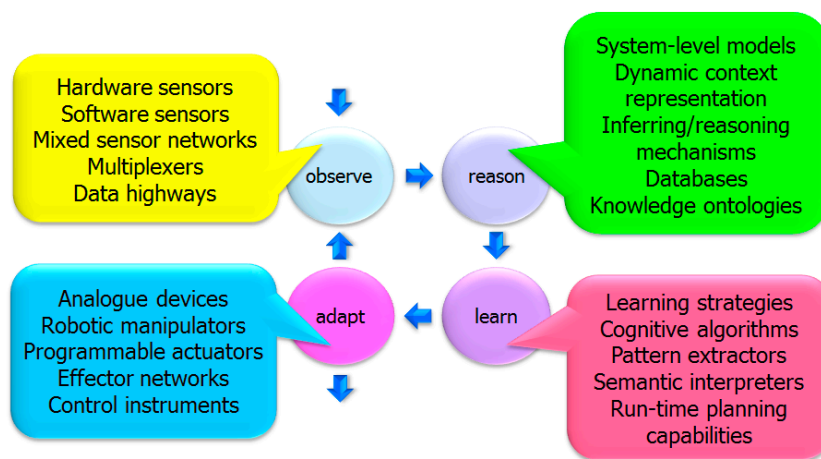


Figure 1. Four clusters of enablers for smart cyber-physical systems.

The understanding of the fundamentals and affordances of CPSs is more nuanced today than it was a decade ago. Therefore, new interpretations and definitions have been proposed. For instance, smartness has been put into a historical dimension and interpreted as a given stage of progression with regards to having cognitive competences and operationalization of a body of knowledge (without considering possible socialized collective intelligence of a system of systems yet). It has been identified as a distinguishing paradigmatic feature of next generation CPSs [8]. Equipping systems with cognitive capabilities, i.e., the process of intellectualization, is the major concern of cognitive engineering of CPSs. However, this domain of interest is still in a premature stage. The guiding assumption is that smart systems should be able to: (i) operate according to dynamically varying, even undefined, circumstances and control regimes, (ii) build awareness of the operational state of their entirety, components, and embedding environment, and (iii) adapt themselves in order to achieve the best possible operational objectives and performance. In other words, smart operation and servicing of CPSs require the capabilities of self-awareness, contextualized reasoning and learning, and functional and architectural morphing [16].

Owing to context-dependent reasoning and system-level adaptation, smart CPSs can be applied in many non-traditional human- and socially-centered applications. However it has also been comprehended that smartness may be utilized in rather different forms in different systems, such as smart cities, homes, transportation, clouds, manufacturing, production, and smart service systems [17], and even smart everything [18]. These systems will eventually show largely diverse levels of intellectualization. Evidently, implementation of smartness is inseparable from the realization of cyber-physical computing (CPC) that intends to go beyond the classical (predefined explicit

algorithms-based) computation that is underpinned by the von Neumann theory of computing [19]. CPC complements or replaces predefined flows of computation with run-time devised flows, and appends run-time constructed or acquired (hidden, but learnable) implicit algorithms to the (preprogrammed) explicit algorithms. The implicit algorithms may manifest as patterned data structures, situated computing strategies, and context-driven reasoning mechanisms.

Obviously, an intense foundational research and a lot of system prototypes-based engineering studies are needed to fully explore and exploit smart system behaviors and affordances [20]. In particular, for the reason that even higher level of intellectualization as smartness is supposed to be present in third- and fourth-generation CPSs. Namely, second-generation CPSs self-generate awareness and perform some-level of functional/architectural self-adaptation under varying conditions [21], whereas third generation CPSs are supposed to have self-cognizance (i.e., awareness with semantic understanding) and non-biological self-evolution capability. As ultimate realizations, fourth-generation CPSs are assumed to have human-resembling system intelligence, which may involve computational consciousness, dependable reasoning, decisional autonomy, and non-genetic self-reproduction [8]. Current systems science and artificial general intelligence (AGI) research are still unable to describe and explain the above-mentioned 'beyond-smartness' types of system behaviors and to inform system engineering about feasible and efficient implementations. At the same time, many recent results validate the concept of second-generation CPSs. The feasibility and utility of specific implementations powered by context management, machine learning and other reasoning mechanisms have been demonstrated.

1.2. On the Background Research and the Research Issue Addressed in This Article

Apprehension of the above-mentioned novel system features calls for a progressive definition, which puts CPSs into the position of self-managing engineered systems. Our definition claims that smart CPSs: (i) employ the principles of cyber-physical computing, (ii) closely interact with the hosting environment(s), (iii) deeply penetrate into physical, biological, social, cognitive, etc. processes, (iv) act as a purposefully arranged set of adaptable actors in these contexts, and (v) possess the capability of non-organic complexification and self-organization. According to this definition, not only the software (middleware) establishes a tight relationship between the physical realm and the cyber realm, but also the mentalware, which is jointly possessed by the concerned system and its stakeholders. This was the starting point and motivation for our background research.

The reported work was conducted in the framework of the portfolio research of the Cyber-Physical Systems Design Section of the Faculty of Industrial Design Engineering at the Delft University of Technology. The shared theme of all related research projects is cognitive engineering of smart cyber-physical systems. The more than ten interrelated projects of staff members and PhD students addressed issues such as: (i) capturing and inferring from dynamic contexts representations, (ii) building system awareness real-time, (iii) computational mechanisms for reasoning, (iv) strategies for functional/architectural adaptation, (v) dynamic system messaging, and (vi) specific applications of smart CPSs. The main research questions for the portfolio research were:

- Based on what theoretical and methodological foundations can system-level smartness be implemented in next generation cyber-physical systems?
- What way can dynamic context information processing be extended to provide semantically enriched awareness representation?
- In what forms can procedural abstraction be implemented as a system-level reasoning mechanism of smart CPSs?
- Based on what knowledge can an active engineering framework support transdisciplinary development of compositional CPSs?
- What do human/system-in-the-loop and supervisory/operative control mean in the context of smart CPSs?

On the one hand, the completed research involved a comprehensive analysis of the state of the art as well as investigation of the pioneering research initiatives. On the other hand, it included a purposeful synthesis of underpinning theories and computational methodologies, and development of testable prototypes.

1.3. Content and Structure of the Article

The most important scientific concerns of the background research were: (i) development of a new reasoning model (a conceptual advancement model) concerning the generations of CPSs based on their level of self-intelligence and self-organization, (ii) implementation of multiple demonstrative prototype (sub)systems in various applications, and (iii) conceptualization of a robust computational framework for procedural abduction (PA) as an ampliative system-level reasoning and adaptation mechanism for S-CPSs. The content of this article contributes to the last concern. PA comprises a contemplation part (that helps realize some level of system self-awareness) and an alteration part (that supports functional and architectural self-adaptation under varying conditions). From an implementation point of view, PA consists of a large set of computational algorithms, which are run-time activated and integrated for reasoning and strategizing.

Though the presented results blend many research aspects of cognitive engineering of cyber-physical systems, only the functional and architectural specification of the PA framework is discussed in the rest of the article. The Section 2 analyses the current state of the art, considering five perspectives: (i) theoretical understanding of mind-like behavior of artefactual systems, (ii) potential enabling technologies for smart CPSs, (iii) achieving system-level holism in reasoning and decision making, (iv) the logical basis of system level reasoning, and (v) recent efforts to exploit abduction as an ampliative computational mechanism. The Section 3 provides a brief overview of the four prototype systems that hinted at the necessary constituents of procedural abduction as a system level reasoning mechanism. It discusses the experimental implementation of the systems for: (i) system-level features composition and compositional system modeling, (ii) identification and forecasting failures in a resilient cyber-physical greenhouse, (iii) representation and reasoning with dynamic context knowledge in a fire evacuation aiding system, and (iv) reasoning and adaptation in an engagement monitoring and enhancing system for stroke rehabilitation. The issues of synthesizing the constituents of the PA mechanism are also considered. The Section 4 presents the computational framework of procedural abduction. It provides a formal specification of the general workflow and the constituents, the underpinning knowledge, and the required functionality of the enabling algorithms of PA. The Section 5: (i) reflects on the completed work, (ii) states some vital propositions, and (iii) makes an inventory of the immediate and future research opportunities.

2. Current State of the Art

2.1. Theoretical Understanding of Mind-Like Behavior of Artefactual Systems

Thinking about transferring human-like intelligence to artificial structures has a long tradition. As known, this process commenced with computerization and, through informatization and cyberization, reached the stage of intellectualization of engineered systems. The development of software science/engineering and information/knowledge engineering proved to be the major driver of the progress. Using the terms and the argumentation of D.M. MacKay, the essence is in the move from 'slave-machines' to 'actor-machines' [1]. As actor-machines, intellectualized artefactual systems are not alternatives of current digital computing systems, but something else and more. Mentioned by the above author, the most fundamental question is "Can an artefact be made to show the behavioral characteristics of an organism?" MacKay argued that this kind of systems should be able to perform important functions such as (i) receiving, selecting, storing, and sending information, (ii) reacting to changes in their 'universe', including data on their own state, (iii) reasoning deductively from premises which are results of previous deductions and data on the different courses, (iv) observing and

controlling their own activities, or otherwise, so as to further some goal, and (v) changing their own pattern of behavior so as to develop quite complex and superficial characteristics capable of rational description in purposive terms.

If we systematically replace the above phrases with our modern terms such as (i) networked sensing, (ii) awareness building, (iii) situated reasoning, (iv) self-organization, and (v) adaptation in context, then we can conclude that D.M. MacKay eventually described the most important paradigmatic features of smart cyber-physical systems. It is a surprising fact since the description happened at the very beginning of 1950s. In addition to stating that future engineered systems are subjects of a reasonably high level of intellectualization and socialization, feature self-planned and goal-directed activities, and potent to be self-sustaining, MacKay also called the attention to the need of sophisticated system-level reasoning mechanisms. He proposed a probabilistic reasoning-mechanism as a possible implementation of it. In addition, like many others, he also emphasized the importance and necessity of the capability of abstraction (by an intellectualized system) [22].

Fifty years later, Russell, S. and Norvig, P. proposed the following taxonomy of systems with non-natural intelligence: (i) systems that think like humans (e.g., cognitive architectures and neural networks), (ii) systems that act like humans (e.g., natural language processing, knowledge representation, learning, and automated reasoning systems), (iii) systems that think rationally (e.g., logic solvers, inference, and optimization systems), and (iv) systems that act rationally (e.g., software agents and embodied robots with perception, planning, reasoning, learning, communicating, and decision-making capabilities) [23]. The above taxonomy indicates the versatility of current systems with non-natural intelligence. They typically require a correct and complete model of the problem as well as of the application domain. However, in the case of a large problem or domain, construction of such models is either rather difficult or not possible at all. Furthermore, if the intellectualized system or its environment, or the set of tasks and the contexts of problem solving frequently and dynamically change, then a steady problem model and application domain model may become inadequate or even inappropriate [24]. In this case, run-time adaptation or complementation of the system model is needed.

2.2. Potential Enabling Technologies for Smart CPSs

In the last sixty years, many efforts were made to develop formal reasoning and learning mechanisms and to implement AI-based systems. Concerning the set objectives, we differentiated: (i) ambitious (striving for ultimate intelligence and autonomy), (ii) realist (constructively exploiting technologies), and (iii) modest (underestimating the affordances and potentials) visions of AI research. Domingos, P. identified five approaches to creating AI-based systems, namely: (i) “symbolist”, implementing logical reasoning based on abstract symbols, (ii) “connectionist”, building structures inspired by the human brain, (iii) “evolutionist”, using methods inspired by Darwinian evolution, (iv) “Bayesians”, using probabilistic inference, and (v) “analogizer”, extrapolating from similar cases seen previously [25]. Within each approach, a large number of different realizations can be found. For instance, symbolist/analyst approaches include (i) standard learning algorithm for monomials, (ii) rule-based inductive algorithms, (iii) instance/pattern/evidence-based algorithms, (iv) search space/decision tree-based algorithms, and (v) probabilistic/fuzzy/non-monotonic algorithms. The group of sub-symbolic approaches includes, among other, (i) genetic algorithms, (ii) neural network algorithms, (iii) support vector machines, (iv) naive Bayesian-classifier algorithms, (v) Bayesian-learning algorithms, and (iv) extreme learning machines. As complements of purely symbolic and sub-symbolic approaches, composite approaches are also often considered. One of the main issues is reasoning with incomplete knowledge [26]. In addition, as discussed by Reed, S.K. and Pease, A.; practically all systems confront obstacles when reasoning needs to be done based on imperfect knowledge (consisting of ambiguous, conditional, contradictory, fragmented, inert, misclassified, or uncertain parts) [27].

Several papers have recently been published on the implementation of various levels of human intellectual/cognitive behavior in consumer durable-type of products, as well as in cyber-physical systems [28,29]. Though some 'wicked-problems' have been considered, mainstream AI research focuses on problem solving means for reasonably well-defined problems. Based on this, it became known that two-valued logical reasoning techniques are not sufficient in the case of systems working with uncertainty or multiple choices [30]. However, the major issue is that integration of problem solving means is complicated since they usually rely on different information/knowledge representation schemata and are implemented by non-compatible algorithms. Among the limited number of related integrative works, the effort of Lees, B. to combine a rule-based reasoning mechanism with a case-based reasoning mechanism is worth paying attention, since it casts light on the challenge of co-evolution of the different bodies of knowledge and the mechanisms themselves [31]. Lumer, C. and Dove, I.J. pointed at the fact that some schemes of reasoning proposed in the literature, including abduction, has been found defective due to the lack of an epistemological backing and, in most cases, the inability to differentiate various degrees of uncertain justification [32].

Though the latest developments of computational learning (CL), machine learning (ML) and deep learning (DL) gave an impetus for smart systems development, the need for explanatory learning has also been identified [33]. This is still hardly supported by robust algorithms of near-zero learning time. ML is mainly concerned with deriving rules, patterns or procedures that explain a body of data or predict future data typically based on statistical processes [34]. The approaches are typically sorted into three categories: (i) supervised learning (based on decision trees, support vector machines, production rules inference, artificial neural network, genetic algorithms, game theory driven approaches, neural network, etc.), (ii) unsupervised learning (generalized additive statistical methods, tree-based methods Bayesian non-parametric approaches, semi-supervised clustering, etc.), and (iii) reinforcement learning (model-free reinforcement learning algorithms, genetic algorithms/programming, feudal Q-learning, adaptive heuristic critic, transfer learning methods, multi-agent reinforcement learning, real-time dynamic programming, etc.). DL typically uses structures of large number of processing layers loosely inspired by the human brain to explore practically the same [35]. Many of these learning technologies have reached a high level of sophistication and increased the potentials of context-independent problem solving [36]. For instance, extreme learning machines combine conventional artificial learning techniques and biological learning mechanism into a suite of machine learning techniques for feedforward neural networks with single and multiple hidden layers, in which hidden neurons need not be tuned [37]. These and other emergent learning techniques aim at computational learning approaches for advanced industrial systems, such as cyber-physical production systems.

2.3. Achieving System-Level Holism in Reasoning and Decision Making

System scientists claim that smartness, like safety, reliability, awareness, adaptability, security, etc., is an overall paradigmatic feature of the systems as a whole, and is not a behavioral characteristic of their specific components, though their functionalities are needed for the realization. This means, smartness is not a reductionist property and cannot be realized by a simple composition of the operations of some smartness-enabling, interface-able components. Furthermore, system smartness cannot be regarded just as a momentary operation of the intellectual mechanisms of a given system, but it should be considered as an always present (ubiquitous and lasting) set of capabilities (with which the system has been equipped with by its developers, or what the system itself develops based on the preprogrammed or the learnt/aggregated knowledge). Owing to learning, system smartness may evolve over the life of a system. Thus, system learning is to be seen as a sustained and combined cognitive process of knowledge acquisition and behavior adaptation.

According to Finocchiaro, M.A., reasoning is the activity of the human mind that consists of reaching conclusions on the basis of reasons, giving reasons for conclusions, and/or drawing consequences from premises [38]. The arguments created for reasoning by human mind are linguistically expressed and arranged based on others. A quasi-automated generation and

representation of arguments or decision points are seen as the major challenge for computational reasoning. In addition, recognizing the sufficiency of argumentation and maintenance of the validity of the arguments are additional challenges [39]. In particular generating informal proofs that are not directly driven by logical rules or physical causalities is problematic [40]. Informal arguments typically depend on their logical form as well as on their content and contexts. Informal proves cannot be expressed in a general logical language (i.e., by explicitly defined well-formed formulae), and cannot be applied successively on explicitly specified logical inference rules or axioms [41].

In the context of system-level reasoning, holism can be addressed properly only if the reasoning strategies, reasoning mechanisms, and reasoning algorithms are simultaneously considered. Reasoning strategies establish the logical and/semantic framework of reasoning. Reasoning mechanisms are implementations of the strategies as computational processes. Reasoning algorithms are active elements used in the holistic reasoning process. Håkansson, A. et al. identified eight strategies of reasoning as dominant ones for smart CPSs, namely: (i) deductive, (ii) inductive, (iii) abductive, (iv) analogical, (v) common sense, (vi) non-monotonic, (vii) case-based, and (viii) probabilistic reasoning strategies [42]. The three enablers (strategies, mechanisms, and algorithms) may be: (i) provided for, (ii) modified by, and/or (iii) developed by a smart CPS. If one strategy is not sufficient enough, then a purposeful mix of different strategies can be considered. This raises the issue of process, algorithm and data integration. The majority of the algorithms known from AI research is typically not interchangeable and cannot be integrated directly into comprehensive reasoning mechanisms. Towards this end, the standard reasoning algorithms included in a complex reasoning mechanism must be adapted to the other related algorithms in the design time. Like customization of the mechanisms to applications, adaptation of algorithms run time by the system itself is a complicated task.

2.4. Possible Logical Bases of System-Level Ampliative Reasoning Mechanisms

A starting point of our mechanism synthesis was the differentiation between formal and informal reasoning. Formal reasoning (often also called deductive reasoning or logic) manipulates the statements of evidence by evaluating them in virtue of their sentential structure and content. Deductive inference supports only explicative inference, where the conclusion is explicitly or implicitly included in the premises. Thus, deductive inference is enumerative since it spells out information that is already contained in the given premises [43]. Deductive arguments provide grounds for making their conclusions inescapable. If the premises are true, then the argument is valid for the reason of syllogism. If all premises are true and the argument is valid, then it is sound. Informal reasoning arrives at a conclusion by means of informed guess-work, relying on amplified evidences. Also called inductive reasoning or logic, informal reasoning interprets the evidences in their correlations or context to arrive at a conclusion. As the principle of reasoning, inductive arguments attempt making a conclusion likely or probable by delivering evidences for it. Provided the premises are true and an inductive argument succeeds in this attempt, the argument is strong. If the argument is both strong and has all true premises, then it is regarded as cogent.

In the context of computational reasoning, the term ‘ampliative’ is used in the meaning of ‘extending’ or ‘adding to what is already known’. It refers to the fact that the conclusion of such argument goes beyond, or amplifies upon, the premises. Actually, this type of inferences is called ampliative and the reasoning mechanisms providing these are referred to as ampliative reasoning engines [44]. Ampliative reasoning may produce conclusions that contain genuinely new information. While deductive inference is enumerative, inductive inference is ampliative in the sense that it goes beyond merely spelling out the information already contained in its supporting evidential premises. While merely explicative (analytic) logical reasoning typically does not add anything to the content of cognition, ampliative (synthetic) logical reasoning increases the given cognition. A characteristic of ampliative reasoning is that the conclusions it yields may be mistaken. Thus, an ampliative argument is not deductively valid or invalid.

Though the idea of ampliative reasoning is well known in mathematical logics and artificial intelligence research, the idea of ampliative system-level reasoning mechanisms is new. Both deductive and inductive reasoning strive after rendering the best judgement in a holistic way, i.e., considering all influencing matters. If this is not the case, then the reasoning is restricted to the available evidences and inferring can target only the best explanation. This process of hypothesis formation and inferring has been described as abduction. As the third kind of logical inferential reasoning, abduction proceeds from observational data or events to a set of hypotheses, which best explains or accounts for the data. Abduction resembles induction in that it involves a reasoning process for providing hypotheses that explain the given facts, while induction is used to derive general rules from specific facts [45]. Peirce, C.S. argued that in spite of some resemblance, abduction may not be regarded merely as a variant of induction, because the mental processes involved are sufficiently distinctive [46]. Abduction involves coming up with plausible explanations for existing data, with the possibility of predicting the existence of additional data which, if subsequently discovered in accordance with its predictions, would tend to confirm the validity of the original hypothesis [47]. Some researchers argued that the psychology of abduction is somewhat mysterious, since it requires creative thought and imagination. In other words, it supposes the ability to imagine possible factual scenarios and to concentrate on just those having the greatest salience for the task in hand. On the other hand, abduction has a strongly ampliative character, which comes from hypothesizing. It has to be mentioned that reproduction of human creative thought and innate imagination by a computational reasoning process is a huge challenge that is not coped successfully by current artificial intelligence research.

In the context of smart CPSs, synthetic judgments are supposed to be system-derived elements of reasoning. A synthetic judgment can be ampliative, if its predicate adds something new to its subject. Ampliative reasoning is heuristic in the sense that it involves obtaining new pieces of beliefs, which are not entailed in the given premises (not already implied by what is known by the system and captured in the system operation/servicing model). It depends on the number and relationships of the finite set of evidences available in the process of reasoning, and is limited by the sophistication of the inferring capabilities of the computational reasoning mechanism concerned. Meheus, J. et al. refer to abduction as ampliative adaptive logics [48]. In their view, abduction is non-monotonic (i.e., conclusions derived at some stage of the reasoning process may be modified or even rejected at a later stage). Their research effort concentrated on a proof theory that warrants that the conclusions derived at a given stage are justified in view of the insight in the premises at that stage. This logic-based theory is supposed to support justified propositions even when the premises imply some sort of undecidable conclusions. Treating abduction as a form of forward reasoning was the logical solution for ending up with ampliative adaptive logics [49].

The above analysis suggests that ampliative reasoning cannot be else but a risk-taking strategy when implemented by smart CPSs. Thus, an objective of the development of system-level reasoning and adaptation mechanism is to reduce the risk of educated guess, which always accompanies decision making by systems. Usually, there are no exact criteria, conditions or measures of when the outcome of reasoning is approximately correct or good enough. It may well be that further evidence, which does not affect the truth of the premises, renders the outcome of reasoning false. This is the reason of why some traditional computational reasoning mechanisms of computer science and artificial intelligence, such as non-monotonic logics, probabilistic logics, reasoning by circumscription, and default reasoning, are not considered as fully-fledged implementations of system-level reasoning and adaptation.

2.5. Recent Efforts to Exploit Abduction as an Ampliative Computational Mechanism

Abduction has multiple interpretations and views [45]. Peirce, S.C. described it as a mode of reasoning that justifies beliefs about the probable truth of theories [46]. It is also seen as a recipe for generating new theoretical discoveries as well as a mode of reasoning that justifies beliefs about the probable truth of theories [47]. Other authors understood it as the process of forming an explanatory supposition [48] or a speculative reasoning strategy [49]. It is the only logical operation which

introduces any new idea and a type of hypothesis formation and logical inference akin to guessing [50]. Abduction may have its suppositions in logic or in knowledge [51] [52]. Hermann, M. and Pichler, R. formally described logic-based abduction as follows: Given a logical theory T formalizing an application, a set M of manifestations, and a set H of hypotheses, find an explanation S for M , i.e., a suitable set $S \subseteq H$ such that $T \cup S$ is consistent and logically entails M [53]. In addition to reasoning in science, abduction has also been considered as a reasoning strategy in engineering and design contexts. As early as 1993, Kean, A.C. described a comprehensive framework for a domain independent abductive reasoning system and proposed to separate inference from domain dependent problem solvers in a computational reasoning framework [54]. This leads to domain independent inference engines, which are portable and applicable to many application domains, and reduce the repetitive efforts to build inference engines.

Computational implementation of abduction has been part of artificial intelligence research. Eiter, T. et al. defined a general abduction model for logic programming to allow the user to define the inference operator (i.e., the programming semantics to be applied on programs) [55]. Gottlob et al. showed that identifying explanations for a given set of observations algorithmically is intractable in the case of logically-based abduction [56]. Therefore, they suggested achieving tractability by reducing the underlying clausal theory so as to have a bounded width for the search tree. However, they also found that turning the theoretical principles of tractability into practically efficient algorithms is very problematic. Though several criteria for selection of explanations have been proposed in the literature, as Poole and Rowen discussed in the context of medical reasoning, several of these criteria are conflicting [57]. Though the efforts to consider uncertainty in computational abductive reasoning proved to be useful, the various approaches were restricted in handling complex, multi-component reasoning problems and in terms of the efficiency of knowledge representation [58]. For example, when used in a smart CPS, one limitation of the Bayesian belief network is that it requires the generation of a network and instantiation of the nodes to be able to explore the posterior probability by some methods.

With regards to computability of abduction, Psillos, S. derived two conjectures: (i) the reasoning process underlying abduction has a certain logical, though not algorithmic, structure, and (ii) the more conceptually adequate a model of abduction becomes, the less tractable it is computationally [59]. The latter finding puts conceptual richness and computational tractability into juxtapositions. He also argued that a rich conceptual model of abduction cannot be adequately programmed. Some major attempts to provide computational models of abductive reasoning are as follows: Pagnucco, M. and Foo, N. proposed an approach to computational abduction based on clausal conceptual graphs, and pointed at some limitations originating, e.g., in the influences of the syntactic restrictions of the graphs and the lack of criteria for selecting the best abduction from among those derived [60]. The abduction model presented by Boutilier, C. and Becher, V. non-monotonically generates explanations that predict an observation, but require some deductive relationship between explanation and observation [61]. Kakas, A.C. et al. used abductive logic programming to develop an abductive reasoning system, called A-System, for declarative problem solving. This involves a hybrid computational model that implements the abductive search in (i) the process of reducing the high level logical representation to a lower-level constraint store, and (ii) a lower-level constraint solving process [62]. Verdoolaege, S. et al. proposed a framework for consideration of temporal information in abductive reasoning in natural language processing, which cannot however be applied directly in the context of CPSs [63].

2.6. Synthesis of the Major Findings

Several theories and technologies such as logic, probability, complexity, physical, biological, cognitive, social theories and technologies, and their various combinations have been developed to realize smart system operations. Direct integration of a bunch of technologies and algorithms does not seem to offer a solution [64]. Since smart systems are supposed to make decisions about their operations and services under varying conditions, they should be able to make conditional inference. Though probability calculus offers formulas for binomial conditional deduction, they are restricted

to dealing with the measure of logical probability. They do not capture the meaning of conditioning and the interplay of multiple conditions [65]. However, conditional reasoning offers a principle for it [66]. Technically, conditional reasoning expresses conditional relationships between parent and child propositions, and then combines those conditionals with evidence about the parent propositions in order to conclude about the child propositions.

Abduction was claimed as a powerful ampliative computational mechanism by many researchers [67]. Abduction was also considered as a logical model of designing. What is to be explained in design is the properness of the overall design objective or goal, and the assumptions are the building blocks of the designs (artefacts) to be synthesized. The explanation is the design (artefact), which is based on the background professional knowledge as well as on the dynamic knowledge associated with design analysis and synthesis. Consistency of reasoning means that designing was possible and that the design (artefact) provably achieved the design goal [68]. Computational complexity was discussed as a challenging issue of abduction [69].

3. Pilot Systems Hinting at Necessary Constituents of Procedural Abduction

3.1. Forerunning Projects and their Outcomes

The idea of procedural abduction was stimulated by the results of forerunning PhD research projects, which investigated various issues of using smart functionalities in second-generation CPSs. They developed different testable pilot implementations for real life scenarios. Figure 2 shows the relationship and contribution of these promotion projects. Their foci were: (i) system-level feature-based conceptualization and integrated operational and architectural modeling of first generation cyber-physical systems [70], (ii) exploring the role of system-initiated changes of operation modes in failure analysis in the context of the above family of CPSs [71], (iii) developing computational mechanisms for dynamic context information representation and inferring, context-based strategy and action planning, and situated messaging in a second generation CPS [72], and (iv) development of a smart reasoning mechanism, and adaptation and intervention planning for a second generation stroke rehabilitation monitoring and enhancing cyber-physical system [73].

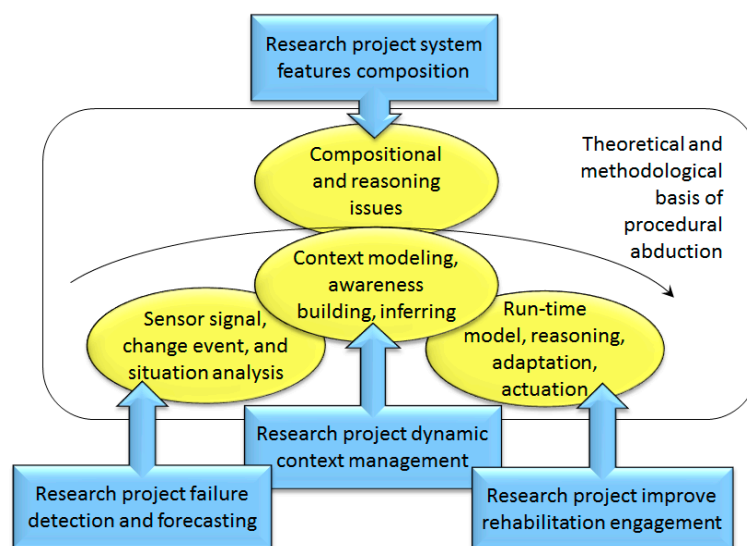


Figure 2. The interplay of the completed PhD research projects in the inception of the concept of procedural abduction.

A common assumption of the above studies is that S-CPSs partially self-determine their operational objectives and system-level adaptation based on run-time collected data, using built-in or acquired smart learning and reasoning algorithms. Relying on multiple streams of input data,

the prototype systems build dynamic operational models, which in turn allow them to alter their operational objectives and to adapt their functionality and (software) architecture accordingly. In the order of mention, these studies cast light on the elements (computational activities) of a complex system-level reasoning process, which reflects the logic and characteristics of abductive reasoning. Consequently, this reasoning process has been called ‘procedural abduction’ (PA) [74]. The proposed theoretical foundations of PA are being challenged currently through analytic investigations (critical systems thinking) and computational implementation in other running PhD [75] and staff research projects [76]. The relevant work will be concisely summarized below.

3.2. System-Level Feature-Based Conceptualization and Modeling of First Generation Cyber-Physical Systems

The objective of this research project was to develop a software toolbox (SMF-TB) for pre-embodiment design of first-generation CPSs. The novelty of this toolbox is that it (i) tries to cope with the inherent heterogeneity of CPSs (caused by interrelated hardware, software and cyberware constituents) on system-level, (ii) combines operational and architectural modeling through (transdisciplinary) system-level features (SLFs), and (iii) provides effective methodological support for creating SLFs by using genotypes, phenotypes and instances [77]. The two main parts and the respective components of the SMF-TB are shown in Figure 3. Likewise the workflow of modeling using SMFs, the chunks of information needed for defining genotypes and phenotypes of SMFs have been clarified and organized into information schema constructs (ISCs). The proposed ICSs connect the chunks of operation and architecture information in the relational data tables of the warehouse databases. The computational algorithms and the overall system modeling mechanisms, including the feature instantiation-based composition mechanism, have been worked out.

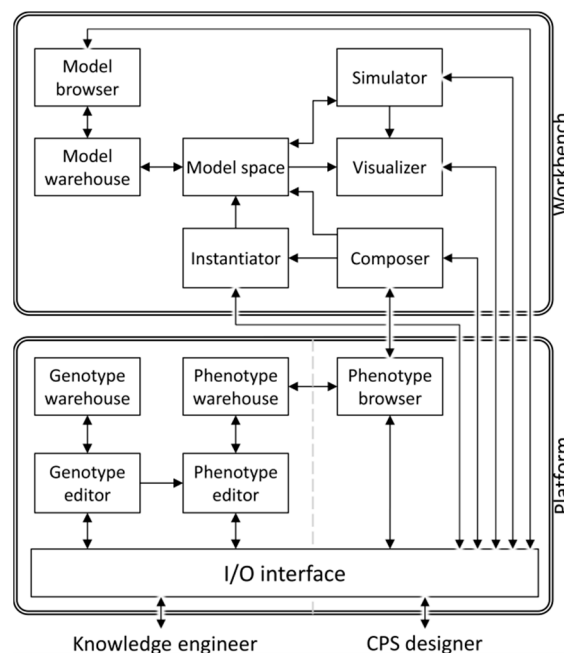


Figure 3. The parts and components of the SMF-TB. (I/O: input/output; CPS: cyber-physical system)

The proposed SMF-TB imposes a strictly physical view, i.e., it models system constituents in the 3D Euclidian space and captures their relationships as mereotopological relations (rather than just representing them by abstract modeling surrogates). The ICSs for: (i) representation of operation and architecture relations, (ii) assigning values to parameter variables, (iii) managing the meta-level knowledge-base of the model warehouse, (iv) recording the composition and parametrization history, and (v) recording the state, event and stream unification history of the instantiation and composition of system constituents are discussed in [78]. The overall process of instantiation involves three cycles,

on genotype, phenotype and instance-type level. Though the proposed information schema constructs, tool-box functionality, system-level modeling methodology, and procedural and computational schemata provide effective support for SMFs-based modeling of first generation CPSs, the completed investigations cast light on various limitations and/or deficiencies with a view to context capturing and self-awareness building, self-reasoning and self-learning, and self-adaptation and self-organization capabilities of smart systems.

The proposed resources proved to be suitable for designing composable systems with definitive interfaces that fulfill input assumption and output guarantee specifications. These conventions however cannot be applied to run-time organized smart systems, which construct their reasoning capabilities during operation and acquire knowledge from run-time processed data streams. Consequently, this project revealed that reasoning mechanisms cannot be synthesized in a components-based manner, that is, by directly combining existing AI/cognitive algorithms. Only compositional synthesis, relying on a holistic knowledge processing framework, can guarantee that the constituents of an ampliative reasoning mechanism provide a synergistic body of knowledge for achieving the objectives of system operation under dynamically varying circumstances. On the other hand, using highly adaptable or self-adaptive system-level (or mechanism-level) features is a reusable idea to address composability challenges of reasoning mechanisms.

3.3. Identification and Forecasting Failures in a Resilient Cyber-Physical Greenhouse Testbed System

This research started out of the assumption that future CPSs will be characterized by growing self-intelligence and self-organization, and considered that these abilities enable them to maintain their operations according to the set objectives under the effects of various influential factors, such as internal failures or external environmental disturbances. Their control regime will set their operation modes, as well as the values of the operational parameters, to achieve the relative best output even under irregular conditions. However, due to the run-time adjustments, they hinder an early recognition of emergent failures. In other words, the transitions self-enabled by a resilient CPS cause uncertainty and hamper the use of the conventional failure analysis methods.

It was hypothesized that the intensity and the trend of the changes of system operation modes (SOMs) can be used as the basis of the diagnosis, recognition and forecasting of failures in resilient CPSs. To test this hypothesis empirically, and to contribute to the theoretical understanding of the failure recognition and forecasting problem, a self-regulatory and self-tuning testbed greenhouse system was developed [79]. The roles of system initiated compensatory actions and operational mode changes were systematically investigated from the aspects of emergence and proliferation of technical failures in this first generation CPS. However, the influence of functional and structural self-adaptation on failure recognition and forecasting was not studied, since it would assume the implementation of an even more sophisticatedly controlled (i.e., smartly behaving) testbed system.

The novelty of this study was that it focused on system-level behavior, rather than on the behavior of the individual components. The input and output signals/data were interpreted on system level (Figure 4), which made it possible to generate a set of indicators that could inform about the SOMs that were triggered by emerging failures. For instance, certain failures pushed the testbed system into an 'abnormal' operation mode, which involved a combination of component operation modes not typical under regular circumstances. On the other hand, certain SOMs did not occur due to the effect of failures, or specific SOMs emerged that did not occur during regular system operation. Consequently, the main contribution of this work was providing novel means, such as the concept of failure induced operation modes, for monitoring changes of states, events and situations of a quasi-dynamic system [80]. The changes were captured through the observed variations in the frequency and the duration of SOMs—that is, in the system dynamics rather than only by signal deviations. This lent itself to a shift from timed system model-based reasoning about the system operation to a data-driven, run-time conditions-based reasoning. Our investigations disclosed that the above concepts could be reused for event and situation diagnoses in other first-generation CPSs and in

feeding reasoning mechanisms with information higher than component input/output data. We also understood that it still needs further elaboration in the case of self-adaptive second-generation systems.

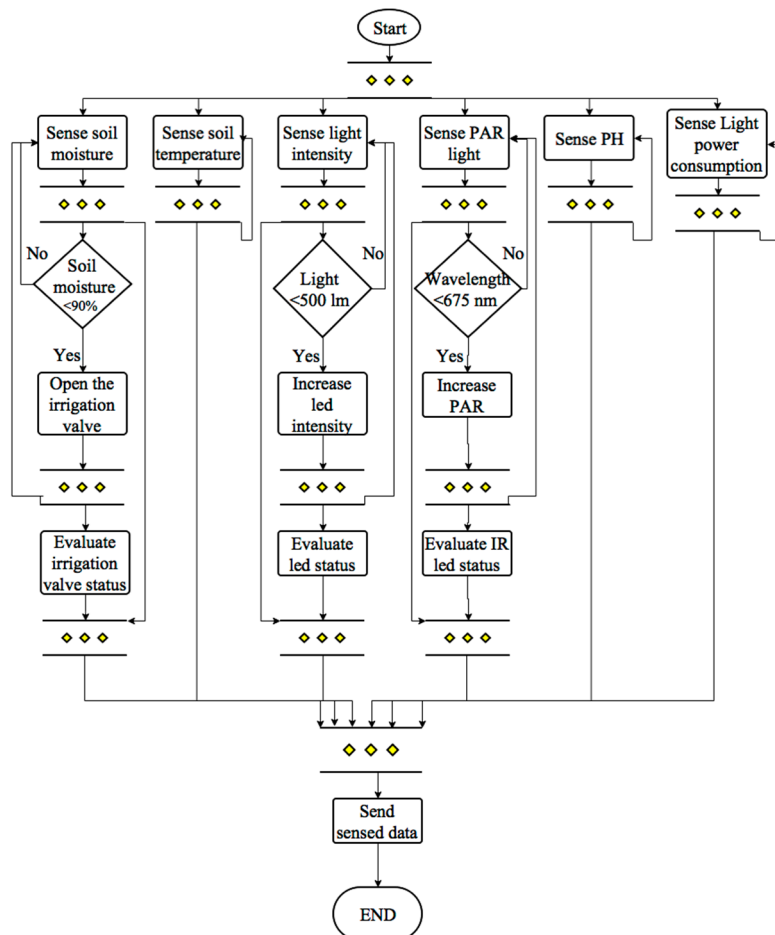


Figure 4. Example of processing sensor information in the testbed greenhouse system. (PAR: photosynthetic active radiation; IR: infrared; PH: scale of acidity).

3.4. Representation and Reasoning with Dynamic Context Knowledge in a Fire Evacuation Aiding System

Smart CPSs often works under highly dynamic circumstances and need to make decisions in dynamic contexts. Typical application examples are such as home care servicing, traffic management on roads, and aiding fire evacuation of buildings. Therefore, it is necessary to provide fast mechanisms for dynamic context computation (DCC) and building awareness by the system that enables it to interpret context changes and to infer about their implications on physical processes. This PhD research project developed a conceptual/logical framework for DCC and implemented computational algorithms for building system awareness. The DCC mechanism was developed based on a layered (context) knowledge model, which included (i) the layer of states (of entities and their spatial, attributive, temporal and semantic (SATC) relationships), (ii) the layer of events (changes in the SATC relationships), (iii) the layer of situations (logical/semantic relations of events and states), and (iv) the layer of scenes (logical/semantic relations of interplaying situations).

The knowledge model assumes information integration on each layer, and information abstraction to support semantic transitions between the subsequent layers. The semantic abstraction over the integrated information constructs provides opportunity to computationally infer not-explicitly described states, events, situations and arrangements in various forms. The data describing dynamic contexts were arranged in a specific computational scheme called context information reference cube (CIR-cube), which proved to be a dexterous computational means for handling spatial, attributive

and temporal data in a cohesive manner (Figure 5) [81]. An inference mechanism was developed that updates and (re)computes the contents of the matrices included in the CIR-cube and eventually builds an awareness model of the time-varying process at hand. The CIR-cube is able to capture both physical time and computational time. The procedure of awareness building includes the following main computation steps: (i) determining the state of the concerned entities, (ii) recognition of events, (iii) identification possible situations, (iv) judging the relevance of situations to the concerned entities, (v) revealing the interplays of the relevant situations, and (vi) interpreting the implications of the interplaying situations [82]. The functional scheme of the inferring mechanism for building awareness is shown in Figure 6.

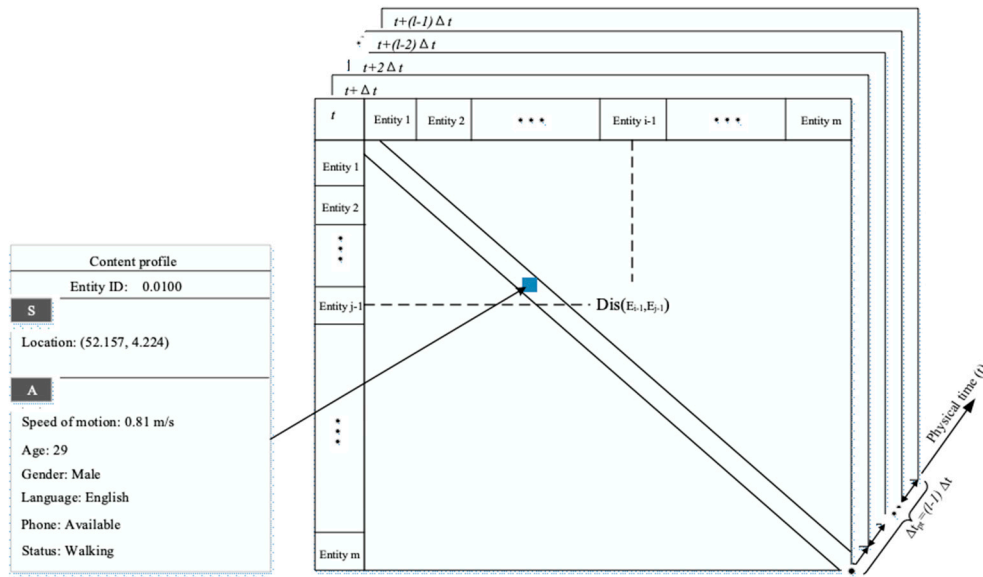


Figure 5. The context information reference cube (CIR-cube).

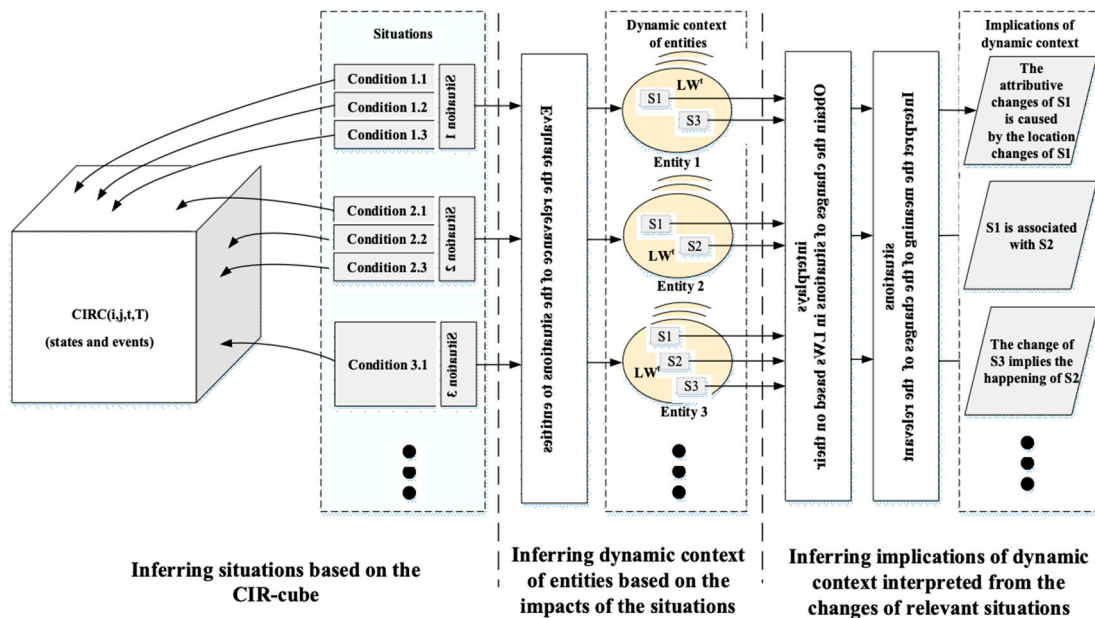


Figure 6. The functional scheme of the inferring mechanism for building awareness. (LW: local world; S: situation).

The developed dynamic context computation mechanism is complemented with a reasoning mechanism for operational strategy synthesis. Ultimately, this mechanism generates personalized

action plans that make it possible to achieve the specific operational/servicing objective of a smart CPS. Together with the implemented additional computational mechanisms for message generation and distribution, the dynamic context computation and action plan generation mechanisms were tested in an indoor fire evacuation guiding application. The application case was investigated based on a high-fidelity simulation of presumed real-life fire propagation and of the behaviors of the human, artefactual, and natural stakeholders. The experimental results proved the efficiency of the interoperating computational mechanisms and algorithms. They also confirmed our hypothesis that the proposed dynamic context computation mechanism was able to provide descriptive knowledge about emergent situations as well as about the implications of the interplaying situations on the concerned entities.

The contribution of this research to the conceptualization of the main constituents of procedural abduction is as follows: The proposed solution uses sensor-provided and/or preprogrammed spatial, attributive and temporal data to describe all involved entities and to characterize their varying states in their local worlds, as a system of reference. A state was defined as a static representation of the momentarily characteristics of an entity, whereas an event as the change of the states of an entity at two subsequent points in time (in the computational realm). A situation is generated by the aggregation of a series of operationally non-independent events and/or states appearing at a given point in time. In addition to computationally combining/integrating logically related events and states, the mechanisms were also able to extract specific meaning of combined and/or integrated events and states by abstraction towards a higher level comprehension. The descriptive data-based context representation is supplemented with various derived elements of semantic intelligence, which are generated by the inferring algorithms of the proposed dynamic context management mechanism.

3.5. Reasoning and Adaptation in an Engagement Monitoring and Enhancing System for Stroke Rehabilitation

Task-oriented training exercises need to be practiced in upper limb rehabilitation after stroke. The hypothesis of this work was that a smart cyber-physical stroke rehabilitation system (CP-SRS), which is able to increase the motor, perceptive, cognitive, and emotional involvement of patients during rehabilitation exercises can be a promising solution. Thus, the objective of this PhD research project was to augment a robot-assisted upper limb rehabilitation subsystem with a cyber-physical computation-based engagement management subsystem. Since there is no opportunity for preprogramming in this specific application case, realization of engagement management raised the need for run time reasoning capabilities. At the beginning of the project it was not completely understood which factors (e.g., game difficulty, personal interest, game design, and immersiveness of environment) are the most influential on the engagement of patients [83]. Furthermore, no quantitative method was found to evaluate momentarily engagement. Based on the outcome of the explorative research, the main functions and architectural elements of the CP-SRS have been defined as: (i) a multi-modal sensor network, which monitors the states of patients, (ii) real time information processing, which interprets the actual signals and generates engagement models, (iii) reasoning and decision making, which provides personalized stimulation plans for different patients, and (iv) situated learning that facilitates run time generation of a reasoning model concerning the system state/objectives and the necessary/possible adaptation. These functionalities and system components were conceptualized and implemented in the CP-SRS at a testable prototype level [84].

Architecturally, the CP-SRS included five subsystems: (i) an assistive robotic subsystem, (ii) a gamification subsystem, (iii) an engagement monitoring subsystem, (iv) a smart learning mechanism (SLM), and (v) an engagement enhancement subsystem (EES) The indicators for: (i) motor engagement (ME), (ii) perceptive engagement (PE), (iii) cognitive engagement (CE), and (iv) emotional engagement (EE) were monitored using: (i) MYO^(TM) Armband with electromyography sensors, (ii) Eyetribe eye tracking device, (iii) Emotiv^(TM) Epoc headset (14 channel wireless electroencephalography (EEG) device), and (iv) a web camera, and Insight device, respectively. The data from these sensors were streamed to MATLAB via TCP/IP computer network transmission control protocol, where the

engagement levels in the four aspects were interpreted. The workflow of information processing related to the smart learning mechanism is shown in Figure 7. Basically, when the patient’s engagement level decreased, the system introduced interventions. Through the interventions it was able to re-engage the patients and to maintain their high level engagement. The interventions meant stimulations in motor, perceptive, cognitive, and emotional aspects, depending on the actual state of the patients. Stimulation strategies were created as a combination of stimulations in multiple aspects.

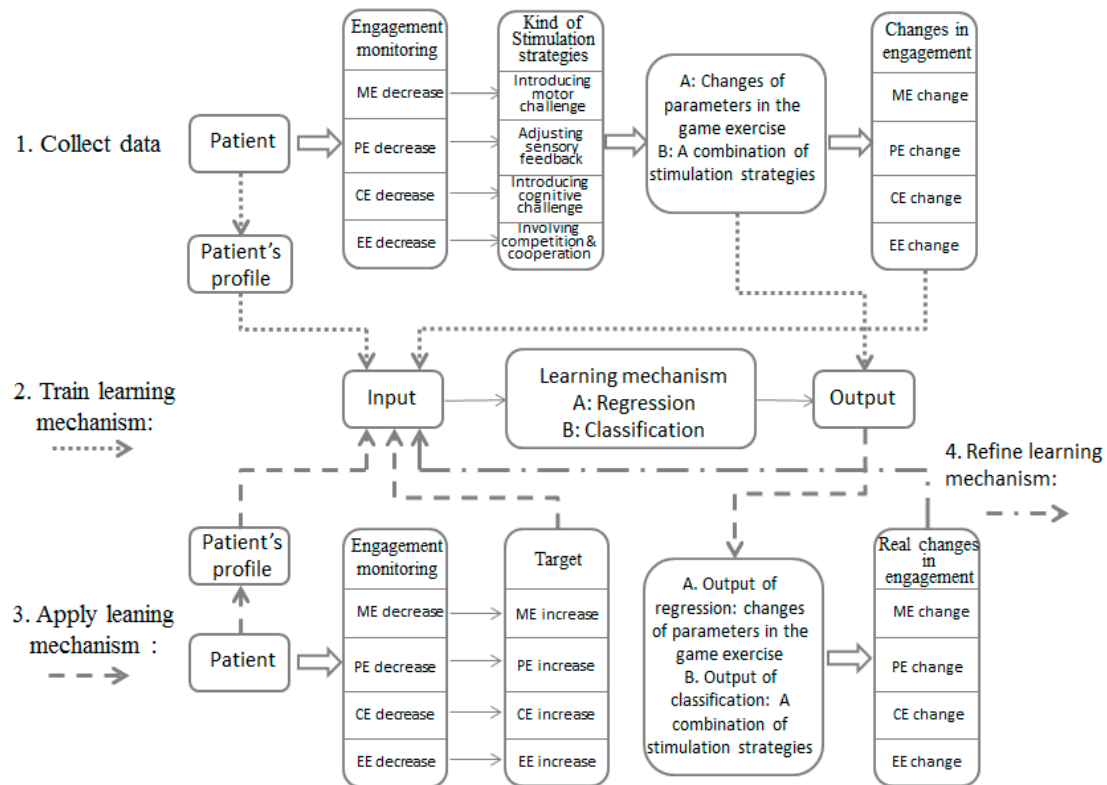


Figure 7. The overall information processing workflow including the smart learning mechanism.

The main findings of this research project can be summarized as follows: It has been found that the ratio of the root mean square of the measured electromyography (EMG) signal and the velocity of motion of the human limb was a reliable indicator of motor (function) engagement. However, the indicators introduced for measuring the motor, perceptive, cognitive and emotional engagements had to be taken into consideration simultaneously in order to achieve an optimal stimulation strategy. They had also to be interrelated in order to form a distinct measure. In the process of computing the applicable stimulation strategies, there was a need to consider the personal profiles of the patients in addition to their motor, perceptive, cognitive, and emotional engagement indicators. A neural network-based smart learning mechanism could be used to learn the effects of the different stimulations strategies on different persons and to propose personalized enhancement. Continuous monitoring of the state of the patient and learning the enhancement options lead to efficient, personalized and self-adaptive stroke rehabilitation training.

The identified engagement indicators were useful not only for enhancing engagement, but also to understand the limitations of the current engagement enhancing methods. Although the methodology developed for monitoring and enhancing engagement was dedicated to rehabilitation, this approach can be used in other fields as well, such as sports, driving, and education. The contribution of this project to revealing constituents necessary for procedural abstraction were: (i) building situation awareness based on input sensor data, (ii) devising a reasoning model run-time based on relative changes of state indicators, (iii) development of possible adaptation (stimulation) strategies by machine

learning using the state indicator-based reasoning model, and (iv) operationalization of the adaptation plan based on changing the setting of the physical and computational effectors.

4. The Framework of Computational Implementation of Procedural Abduction

4.1. The General Workflow and the Underpinning Knowledge of Procedural Abduction

From a computational point of view, procedural abduction is seen as a recurrent sequence of eight processing stages: (i) run-time extraction of data/signals by sensing, (ii) recognition of change events, (iii) inferring about exiting operational situations, (iv) building awareness of the system’s performance, (v) devising alternative performance enhancement strategies, (vi) designing adaptation of the system as a whole, (vii) planning the implied interventions, and (viii) actuating effectors and controls. The operational workflow (computational implementation) of procedural abduction (PA) is graphically shown in Figure 8. The activities (i)–(iv) enable the system to capture data about its momentary performance (outcome of system operation), to compare the data with those representing the assumable best objective of servicing, and to define the necessary and possible changes of the system. The activities (v)–(viii) enable the system to determine the necessary functional and/or architectural changes, to define the adaptations to be introduced, and to set the values of the actuator control variables accordingly. Each activity involves at least one, but typically multiple interacting computational algorithms, which are integrated into the overall reasoning mechanism of procedural abduction.

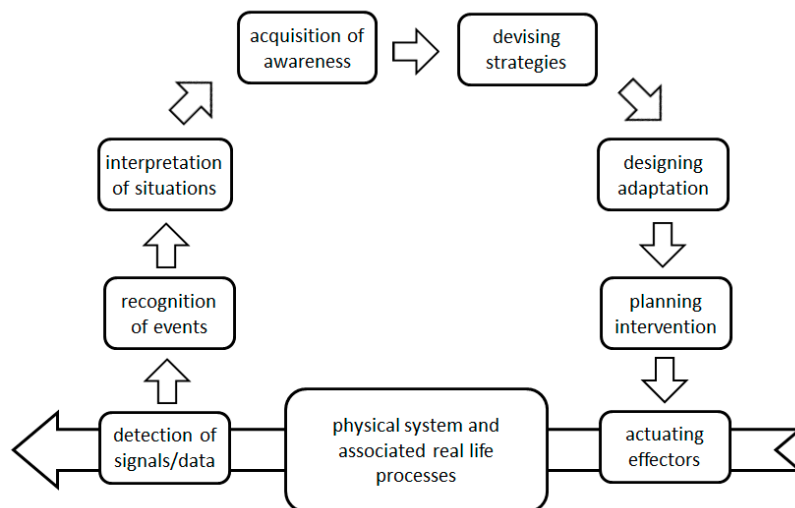


Figure 8. Graphical representation of the generic workflow of procedural abduction.

The basis of implementation of the computational workflow of PA is the predefined operation and servicing model (OSM) of the concerned CPS system, which specifies: (i) the default objectives, (ii) the values of the operational parameters, and (iii) the initial state of the system, as a reference. Symbolically, the OSM can be specified on a given level of resolutions as a septuplet (Equation (1)):

$$OSM = (IV, AC, PS, OD, AD, OV, TV) \tag{1}$$

where: IV is a finite non-empty set of input variables belonging to the total of interoperating system actors on a given level of resolution; AC is a finite non-empty set of architectural constituents realizing the system actors on a given level of aggregation; PS is a finite non-empty set of services generated by the system actors, OD is a finite non-empty set of operational dependences among system actors and the provided services; AD is a finite non-empty set of architectural dependences among system actors and the provided services; OV is a finite non-empty set of output variables describing the provided services, and TV is a finite non-empty set of the required/possible target (interval) values of operation

and services of the system as a whole. An explicit incorporation of the operational and architectural dependences of the system actors is needed in view of the compositional synthesis of the system. (Compositional synthesis means that the operational and architectural manifestation of the system actors as constituents is simultaneously defined by the overall operational/servicing objectives of the system as a whole and by the manifestation of the closely interoperating other constituents.)

PA is not purely a logical propositions-based reasoning, but data-, information- and knowledge-based. The knowledge required for the implementation of the generic workflow of procedural abduction (shown in Figure 8) has two parts: (i) static knowledge and (ii) dynamic knowledge. The static knowledge is conveyed by the predefined OSM of the system (referred to as $OSM_{initial}$ in the rest of the article). The dynamic knowledge is derived based on the run-time acquired data and the modified OSM (OSM_{actual}), and thus by the change of the operational context and objectives. The concerned algorithms of the computational mechanism blend the parts of these two bodies of knowledge in each stages of processing. The overall scheme of using the knowledge sources is shown in Figure 9. From the point of view of realization logical/semantic reasoning by PA, there are four sources of knowledge considered: (i) the content of $OSM_{initial}$, (ii) the content of OSM_{actual} , (iii) the descriptive spatial, attributive and temporal data, semantic relations, and prescriptive constraints included in the permanent context ($\hat{P}\hat{C}$) and dynamic context ($\hat{D}\hat{C}$) representations, and (iv) specifications of conceptualizations in the associated system-used ontologies and data residing on the Web. The necessity of these sources of knowledge has been proven in our research, as well as why their sufficiency for a wide range of applications should still be tested experimentally.

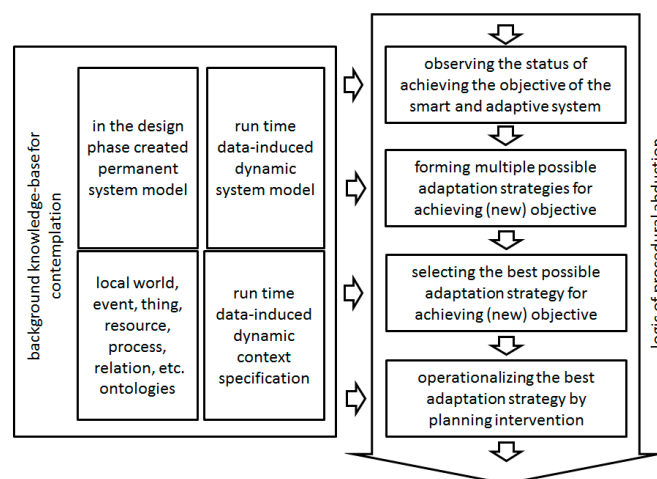


Figure 9. The knowledge assets used in procedural abduction.

4.2. From Enabling Operators to Transforming Algorithms

The computational mechanism of procedural abduction (MPA) has been interpreted as an arrangement of operators completing the computational activities in each stage of PA. MPA can symbolically be represented by the following formula (Equation (2)):

$$MPA = \overset{DD}{\rightarrow} \gg \overset{RE}{\rightarrow} \gg \frac{\overset{FI}{\rightarrow} \gg \overset{AP}{\rightarrow}}{\underset{IS}{\rightarrow} \gg \underset{AA}{\rightarrow} \gg \underset{DS}{\rightarrow} \gg \underset{DA}{\rightarrow}} \gg \overset{PI}{\rightarrow} \gg \overset{AE}{\rightarrow} \quad (2)$$

where: \rightarrow (rightward arrow) indicates an operator of the workflow; (horizontal line) is used to separate alternative sequences of operators (i.e., operation with or without adaptation). The symbol ' \gg ' indicates the orientation of the flow of information processing. The group of operators above the horizontal line allows self-adjustment of a system, while the group of operators below the horizontal line enables self-adaptation of a system. DD is the operator of detecting signals and elicitation of data, RE is the operator of recognizing and monitoring events, FI is the operator of providing

feedback information for operational control; AP is the operator of adjusting the values of operational parameters; IS is the operator of interpretation of situations; AA is the operator of acquisition of operational awareness; DS is the operator of devising adjustment and/or adaptation strategy; DA is the operator of designing adaptation; PI is the operator of planning interventions; and AE is the operator of selective actuating of effectors.

Each element of the computational process, including those belonging to its decision making sub-process, can be implemented by two kinds of computational operators: (i) (physically-grounded) operators handling state changes in the physical realm, and (ii) operators for processing data, information and knowledge in the cyber realm. While the first kind of operators process changes in the continuous space and time, the second kind of operators work with digital representation and feature event-oriented execution. Ultimately, both capture state changes and are handled in similar manner as computational transformations, $T_{x,i}$, where: x is any operator of PA, and i is the identifier of a particular computational action belonging to x . The operators may include three sets of transforming algorithms, namely basic, auxiliary and interaction algorithms. These types are identified based on the purpose of the algorithms. Basic algorithms generate new information by reasoning/infering, while auxiliary algorithms maintain information, for instance, by recording digital data in files. Interfacing algorithms avail information, e.g., receive manual data input, display data to enable human interaction, or convert data between cooperating system modules. Due to space limitation, only the basic transformations (sets of transforming algorithms) will be discussed.

4.3. Transforming Algorithms Needed for the Operators

The operator for detecting signals and elicitation of data (DD) includes the following transformations (Equation (3)):

$$DD = (T_{DD,1}, T_{DD,2}, T_{DD,3}, T_{DD,4}, T_{DD,5}, T_{DD,6}, T_{DD,7}, T_{DD,8}) \quad (3)$$

where: $T_{DD,1}$ is identification of active physical and action sensors in the environment; $T_{DD,2}$ is local sensing of the attributes of material flows; $T_{DD,3}$ is local sensing of the attributes of energy flows; $T_{DD,4}$ is local sensing of the attributes of information flows; $T_{DD,4}$ is collecting data from linked software sensors; $T_{DD,6}$ is transferring signals on the wired/wireless network; $T_{DD,6}$ is multiplexing analogue signals; $T_{DD,7}$ is converting analogue signals into digital data; and $T_{DD,8}$ is cleaning/filtering digital data.

Used for recognizing and monitoring events, the next operator (RE) includes transformation algorithms for the analysis of the sensed and input signals/data in order to detect something that happens or might happen at a given physical or logical place and time. The analysis considers the operational state of the system as a whole, of the constituents, and the set operation/servicing objectives. The recognizing and monitoring events operator is defined as (Equation (4)):

$$RE = (T_{RE,1}, T_{RE,2}, T_{RE,3}, T_{RE,4}, T_{RE,5}, T_{RE,6}, T_{RE,7}) \quad (4)$$

where: $T_{RE,1}$ is detection of change trends in digital data, $T_{RE,2}$ is obtaining information over operation modes of the system, $T_{RE,3}$ is detection of remarkable signal changes that may be associated with discrete change events, $T_{RE,4}$ is features-based investigation of the signal changes, $T_{RE,5}$ identification and classification of a recognized events according to their nature (space-related, attribute-related, or time-related), $T_{RE,6}$ is time stamping and recording of recognized events, and $T_{RE,7}$ is monitoring the life cycle of the recorded events. These transformations make it possible to recognize state changes in terms of deviation from the set objective and the preferred system states, respectively.

A situation has been defined as interactions of the recognized events, not matter if they concern the change in the objectives or in the system-internal states. The main goal of this activity is to identify

the interacting events and to determine their relationships in space, time and logic. Thus, the operator for interpretation of situations (IS) includes the following transformations (Equation (5)):

$$IS = (T_{IS,1}, T_{IS,2}, T_{IS,3}, T_{IS,4}, T_{IS,5}, T_{IS,6}, T_{IS,7}) \tag{5}$$

where: $T_{IS,1}$ is recalling all recognized events; $T_{IS,2}$ is investigation of the space of the individual events in a considered local world based on the location of the signal provider; $T_{IS,3}$ is investigation of the time stamps and durations of the individual events in the considered operation window; $T_{IS,4}$ is computation of spatial relationships of the events occurring in the considered local world; $T_{IS,5}$ is computation of temporal relationships of the events occurring in the considered time window; $T_{IS,6}$ is determining the set of correlated events and recording it as a situation; and $T_{IS,7}$ is monitoring the trend of change of an identified situation.

Having a grasp on the states of system operation and objective achievement, the reasoning mechanism intends to ‘understand’ the meaning and implication of the actual situation. This is based on learning, which allows the knowledge-enabled mechanism to acquire additional information and to build awareness. As discussed above, there are four sources of information: (i) the initial system model, (ii) the run-time acquired data, (iii) the dynamic context model, and (iv) additional data repositories. Building operational awareness allows making logical judgments and inferring conclusions. Thus, the operator for acquisition of operational awareness (AA) is composed of the following transformations (Equation (6)):

$$AA = (T_{AA,1}, T_{AA,2}, T_{AA,3}, T_{AA,4}, T_{AA,5}, T_{AA,6}, T_{AA,7}, T_{AA,8}, T_{AA,9}) \tag{6}$$

where: $T_{AA,1}$ is operationalizing the $OSM_{initial}$; $T_{AA,2}$ is determining the deviations from $OSM_{initial}$ in the given situation; $T_{AA,3}$ is computation of the operation/servicing indicators; $T_{AA,4}$ is generating implicit context information; $T_{AA,5}$ is generating spatial context information; $T_{AA,6}$ is generating temporal context information; $T_{AA,7}$ is generating attributive context information, $T_{AA,8}$ is computation of the dynamic context model; and $T_{AA,9}$ is unsupervised learning of the necessary control regime (that is, if maintaining $O_{initial}$ is needed or if there is a possibility for a more favoring $O_{possible}$). Together with the results of the transformations included in the operators AA and DA, the DS operator informs the reasoning mechanism about how and why a particular set of conclusions were made. That is, the sequence » AA » DS » DA » represents a local proposition-based abduction in the whole process of procedural abduction.

Depending on the control regime, a proper strategy is needed to achieve the set operational/servicing objective of the concerned CPS. This is the task of the devising adjustment and/or adaptation strategy (DS) operator, which includes the following computational transformations (Equation (7)):

$$DS = (T_{DS,1}, T_{DS,2}, T_{DS,3}, T_{DS,4}, T_{DS,5}, T_{DS,6}, T_{DS,7}) \tag{7}$$

where: $T_{DS,1}$ is initiation of computational actions according to the control regime (‘observation’); $T_{DS,2}$ is investigation of operation/servicing indicators with regards to possible enhancement; $T_{DS,3}$ is devising alternative operational strategies; $T_{DS,4}$ is devising feasible associated adaptation strategies (‘hypotheses’); $T_{DS,6}$ is assessing the operational and adaptation strategies (‘stratagems’) considering the resources and the context of actions; and $T_{DS,7}$ is ranking the stratagems and selecting the best one.

While strategizing focusses on both the functional and logical aspects (i.e., what to change and why to change), architecture and operation adaptation concentrates on the technical and practical aspects of altering the system (i.e., on how to change and when to change). In this sense it produces a technical blueprint of the system alteration together with a course adaptation plan. This plan is the basis of intervention specification. The operator for designing adaptation (DA) is realized by the following computational transformations (Equation (8)):

$$DA = (T_{DA,1}, T_{DA,2}, T_{DA,3}, T_{DA,4}, T_{DA,5}, T_{DA,6}, T_{DA,7}, T_{DA,8}, T_{DA,9}) \tag{8}$$

where: $T_{DA,1}$ is investigation of the degrees of freedom in which the system can be adapted according to the best adaptation strategy; $T_{DA,2}$ is determining the necessary/possible operation/servicing adaptation; $T_{DA,3}$ is determining the necessary/possible architecture adaptations; $T_{DA,4}$ is computation of the OSM_{actual} based on the skeleton of $OSM_{initial}$; $T_{DA,5}$ is computational simulation (pre-playing) of the system's operation/servicing after introducing the adaptations; $T_{DA,6}$ is investigation of the impact of adaptation on the system's properties; $T_{DA,7}$ is adjustment of OSM_{actual} according to the findings and enhancement of the adaptation plan; $T_{DA,8}$ is identification of the outgoing and/or incoming system resources; and $T_{DA,9}$ is determining the sequence of the hardware, software and cyberware adaptation actions. As it can be seen, this operator is one of those that require the largest number of interoperating algorithms.

The DA operator can provide only a 'delayed' adaptation plan due to the necessary preliminary computational testing of the impacts of adaptation. The possibility of adaptation is influenced by then completion of certain operations of the CPS. This creates the need for the planning interventions operator. The objective of this operator is to operationalize a refined adaptation plan operation- and time-wise. Actually, it converts the adaptation blueprint into a transition blueprint, which considers the operation/servicing of the CPS and the conditions for this. The operator for planning interventions (PI) comprises the following transformations (Equation (9)):

$$PI = (T_{PI,1}, T_{PI,2}, T_{PI,3}, T_{PI,4}, T_{PI,5}, T_{PI,6}) \quad (9)$$

where: $T_{PI,1}$ is generation of a scenario for modification of the effectors; $T_{PI,2}$ is computation of control information for all motors; $T_{PI,3}$ is computation of the control information for all regulators; $T_{PI,4}$ is computation of the control information for all sensors; and $T_{PI,5}$ is computation of the control information for all information handling components, and $T_{PI,6}$ is computation of the control information for all computational effectors.

Finally, the operator for selective actuation of effectors (AE) is defined as (Equation (10)):

$$AE = (T_{AE,1}, T_{AE,2}, T_{AE,3}, T_{AE,4}, T_{AE,5}) \quad (10)$$

where: $T_{AE,1}$ is activating and setting rotary motors, stepper motors, servos and specialty motors; $T_{AE,2}$ is activating and setting linear actuators, effect transformers, regulators and transceivers; $T_{AE,3}$ is activating and setting environmental sensors, physical sensors and action sensors; $T_{AE,4}$ is activating and setting communicators, transceivers, modems, converters, cameras, and displays setting of system parameters; $T_{AE,5}$ is activating and setting computational effectors.

5. Some Conclusions and Future Research Opportunities

5.1. Reflection on the Approach

Humans typically apply the divide-and-conquer strategy combined with some informal reasoning to solve complex practical application problems. In cyber-physical systems, the complex application problem (e.g., providing multi-activity assistance in home care context) is to be decomposed to tasks that can be allocated to the active nodes of the system. Decomposition of complex problem typically needs informal (intuitive and semantics-based) reasoning. It also assumes certain level of autonomy and collaboration of the active nodes [85]. However, as Pease, A. and Aberdein, A. argued, a comprehensive theory of informal reasoning is not available and perhaps even not expectable [86]. Therefore, problem solving by cognitively enabled systems needs to be based on formal, computationally processable reasoning theories.

The objective of the presented research effort was to contribute to the progress in this field of interest. As a computational implementation of a formal theory that provides a flexible system-level reasoning capability for various smart CPSs, procedural abduction was proposed. Its idea emerged

based on a conceptual synthesis of solutions for reasoning in various application contexts. The reasoning pattern of procedural abduction is straightforward:

Some phenomenon concerning the operation of the system is observed;
If a particular explanation would be true, then the phenomenon could be a matter of course;
Hence, there is a reason to suspect that the particular explanation is proper (true).

Contrary to the above fact, implementation of the computational algorithms and data processing for PA is a complex and challenging undertaking. One of the challenges of implementation is that the computational reasoning should be knowledge-based, rather than just purely logics-based. On the other hand, PA offers affordances and benefits that cannot be expected from other approaches.

In the case of a smart CPS, the observed phenomenon is the relation of the state of the system to the set operational objective, or to a possible optimal operational objective. To generate proper explanations about this relation, the process of reasoning should include a part that collects information and builds awareness about the actual operational state of a system (contemplation), and another part that reasons about the necessary/possible adaptations towards a better operational objective (alteration). These are the functional backbones of the proposed procedural abduction mechanisms. PA exemplifies a form of reasoning that is ampliative in the sense that it aims at extending the domain of the actually existing system knowledge in every given state of its operation.

In principle, PA is application independent, but it should most probably be tailored according to the specificities of the considered application domains in order to achieve the best possible performance. By enabling deep penetration into real life processes and complementing model-based system control regimes with non-preprogrammed, run-time data acquisition-enabled, learning and reasoning, PA may be the reasoning mechanism for many physical, biological, medical, social, cognitive, etc. applications which includes processes of highly dynamically changing nature. Computational realization of PA necessitates a combination of a large number of conventional and specific artificial intelligence algorithms, which should be interconnected in a compositional manner [87]. Achieving compositional synergy in terms of the large number of interdependent algorithms, as well as in terms of the knowledge flow needed for system level problem solving, is found as a challenge for implementation. This problem is already a recognized one in the literature [88].

One of the aims of this article was to emphasize the significance of abduction as a computationally feasible problem solving process and to propose computational framework for procedural abduction. PA operationalizes the principle that systems and agents of cognitive problem solving should incorporate knowledge about the world (ontological commitment) and an abstract procedure (inferential commitment) for interpreting this knowledge towards constructing operation plans and taking informed actions. Clearly, implementation, application and validation of procedural abduction as an ampliative reasoning mechanism for varied cyber-physical systems are a work in progress. However, the development of its underlying theoretical framework and computational methodology has reached an advanced stage. At this time, it can be forecasted that its realization may come to fruition, though a fully-fledged implementation in a form of a platform, which is applicable in multiple CPSs in various contexts, still requires substantial work.

5.2. Future Research Opportunities

The results summarized in this article are related to the first phase of our research, which concentrated on exploring the elements of a feasible conceptual framework for procedural abduction. The on-going research efforts are made towards a fully-fledged computational implementation and integration. The development activities should extend to the refinement of all algorithms chosen or developed for system level reasoning. The ultimate objective is to use the abductive reasoning mechanism as pluggable module of smart CPSs, which can provide application independent reasoning and reduce the software and knowledge engineering work. Since high-fidelity computational replicas of complex mental representations are inherently compositional, conceptual frameworks and design

methodologies fostering compositionality of smart CPSs are highly necessary. Towards this end, the issues associated with compositional design of reasoning mechanisms need to be addressed too. The importance of compositionality is well recognized in the literature, but further research is needed in the case of run time self-organizing systems. The issue of maintaining synergy between the initial system model and the (dynamically changing) actual system model should also be addressed. Even a partial modification of a well-tested system reasoning model by a run time developed reasoning model may create problems with a dependability and resilient operation of CPSs. The uncertainty created by each step of procedural abduction, such as inferring and interpreting the changing context of application, developing adaptation strategies, and performing adaptation of system models, should be treated with utmost care. Further explorative research is needed in this field too. Furthermore, specific methods that are able to verify adjusted system operation models at run-time are also needed. They should be able to investigate and forecast the effects of various operation strategies and system adaptations on the performance and behavior of CPSs in changing contexts. Real time formal verification of operational strategies is an essential feature of procedural abduction. While addressing formal verification at run time is in the focus of CPS research, methods have to be developed that would help address the concomitant challenges.

Funding: This research received no external funding.

Acknowledgments: The author gratefully recognizes the contribution of Shahab Pourtalebi, Santiago Ruiz-Arenas, Yongzhe Li, Chong Li, and Zoltán Rusák to the Cognitive Engineering of Cyber-Physical Systems project.

Conflicts of Interest: The author declares no conflict of interest.

References

1. MacKay, D.M. Mind-like behaviour in artefacts. *Bull. Br. Soc. Hist. Sci.* **1951**, *1*, 164–165. [[CrossRef](#)]
2. Poovendran, R.; Sampigethaya, K.; Gupta, S.K.; Lee, I.; Prasad, K.V.; Corman, D.; Paunicka, J.L. Special issue on cyber-physical systems. *Proc. IEEE* **2012**, *100*, 6–12. [[CrossRef](#)]
3. Rho, S.; Vasilakos, A.V.; Chen, W. Cyber physical systems technologies and applications. *Future Gener. Comput. Syst.* **2016**, *56*, 436–437. [[CrossRef](#)]
4. Knight, J.; Xiang, J.; Sullivan, K. A rigorous definition of cyber-physical systems. In *Trustworthy Cyber-Physical Systems Engineering*; CRC Press: Boca Raton, FL, USA, 2016; Chapter 3; pp. 47–74.
5. Sztipanovits, J.; Koutsoukos, X.; Karsai, G.; Kottenstette, N.; Antsaklis, P.; Gupta, V.; Goodwine, B.; Baras, J.; Wang, S. Toward a science of cyber-physical system integration. *Proc. IEEE* **2012**, *100*, 29–44. [[CrossRef](#)]
6. CPSOS Consortium. *Cyber-Physical Systems of Systems: Research and Innovation Priorities*; Initial Document for Public Consultation; European Union's Seventh Programme for Research, Technological Development and Demonstration; CPSOS Consortium: Düsseldorf, Germany, 2014; pp. 1–21.
7. Schätz, B.; Törngreen, M.; Bensalem, S.; Cengarle, M.V.; Pfeifer, H.; McDermid, J.A.; Passerone, R.; Sangiovanni-Vincentelli, A. Cyber-Physical European Roadmap and Strategy: Research Agenda and Recommendations for Action, CyPhERS Consortium Technical Report. February 2015. Available online: www.cyphers.eu (accessed on 11 October 2018).
8. Horváth, I.; Rusák, Z.; Li, Y. Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems. In Proceedings of the ASME 2017 International Design Engineering Technical Conferences, Cleveland, OH, USA, 6–9 August 2017; pp. 1–14.
9. Sheth, A.; Anantharam, P.; Henson, C. Physical-cyber-social computing: An early 21st century approach. *IEEE Intell. Syst.* **2013**, *28*, 79–82. [[CrossRef](#)]
10. Engell, S.; Paulen, R.; Reniers, M.A.; Sonntag, C.; Thompson, H. Core research and innovation areas in cyber-physical systems of systems. In *Cyber Physical Systems. Design, Modeling, and Evaluation*; Springer International Publishing: Berlin, Germany, 2015; pp. 40–55.
11. Colombo, A.W.; Bangemann, T.; Karnouskos, S.; Delsing, J.; Stluka, P.; Harrison, R.; Jammes, F.; Lastra, J.L. *Industrial Cloud-Based Cyber-Physical Systems. The IMC-AESOP Approach*; Springer Science + Business Media: Berlin/Heidelberg, Germany, 2014; ISBN 3319056239 9783319056234.

12. Yue, X.; Cai, H.; Yan, H.; Zou, C.; Zhou, K. Cloud-assisted industrial cyber-physical systems: An insight. *Microprocess. Microsyst.* **2015**, *39*, 1262–1270. [[CrossRef](#)]
13. Gerritsen, B.H.; Horváth, I. Current drivers and obstacles of synergy in cyber-physical systems design. In Proceedings of the ASME 2012 International Design Engineering Technical Conferences, Chicago, IL, USA, 12–15 August 2012; pp. 1277–1286.
14. Ribeiro, L.; Barata, J. Self-organizing multiagent mechatronic systems in perspective. In Proceedings of the 11th International Conference on Industrial Informatics, IEEE, Bochum, Germany, 29–31 July 2013; pp. 392–397.
15. Palviainen, M.; Mäntyjärvi, J.; Ronkainen, J.; Tuomikoski, M. Towards user-driven cyber-physical systems—Strategies to support user intervention in provisioning of information and capabilities of cyber-physical systems. In *Industrial Internet of Things*; Springer International Publishing: New York, NY, USA, 2017; pp. 575–593.
16. Juuso, E.K. Integration of intelligent systems in development of smart adaptive systems. *Int. J. Approx. Reason.* **2004**, *35*, 307–337. [[CrossRef](#)]
17. Barile, S.; Polese, F. Smart service systems and viable service systems: Applying systems theory to service science. *Serv. Sci.* **2010**, *2*, 21–40. [[CrossRef](#)]
18. Zhuge, H. Cyber physical society. In Proceedings of the Sixth International Conference on Semantics Knowledge and Grid, IEEE, Beijing, China, 1–3 November 2010; pp. 1–8.
19. Hahanov, V.; Litvinova, E.; Chumachenko, S.; Hahanova, A. Cyber physical computing. In *Cyber Physical Computing for IoT-driven Services*; Springer: Cham, Switzerland, 2018; pp. 1–20.
20. Stankovic, J.A. Research directions for cyber physical systems in wireless and mobile healthcare. *ACM Trans. Cyber-Phys. Syst.* **2016**, *1*, 1–12. [[CrossRef](#)]
21. Leitão, P.; Colombo, A.W.; Karnouskos, S. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Comput. Ind.* **2016**, *81*, 11–25. [[CrossRef](#)]
22. Sobhraján, P.; Nikam, S.Y.; Pimpri, D.; Pimpri, P.D. Comparative study of abstraction in cyber physical system. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 466–469.
23. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Englewood Cliffs, NJ, USA, 1995; ISBN 0-13-103805-2.
24. Tani, J. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 421–436. [[CrossRef](#)] [[PubMed](#)]
25. Domingos, P. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*; Basic Books: New York, NY, USA, 2015; pp. 1–315.
26. Jamroga, W.; Ågotnes, T. Constructive knowledge: What agents can achieve under imperfect information. *J. Appl. Non-Class. Log.* **2007**, *17*, 423–475. [[CrossRef](#)]
27. Reed, S.K.; Pease, A. Reasoning from imperfect knowledge. *Cogn. Syst. Res.* **2017**, *41*, 56–72. [[CrossRef](#)]
28. McFarlane, D.; Giannikas, V.; Wong, A.C.; Harrison, M. Product intelligence in industrial control: Theory and practice. *Annu. Rev. Control* **2013**, *37*, 69–88. [[CrossRef](#)]
29. Meyer, G.G.; Främbling, K.; Holmström, J. Intelligent products: A survey. *Comput. Ind.* **2009**, *60*, 137–148. [[CrossRef](#)]
30. Flasiński, M. Reasoning with imperfect knowledge. In *Introduction to Artificial Intelligence*; Springer: Cham, Switzerland, 2016; pp. 175–188.
31. Lees, B. Smart reasoning. In *Artificial Intelligence: Learning Models*; Pearson Education: London, UK, 2013; pp. 1–3.
32. Lumer, C.; Dove, I.J. Argument schemes—An epistemological approach. In Proceedings of the OSSA Conference Archive, Windsor, ON, Canada, 8–21 May 2011; Volume 17, pp. 1–5.
33. Resnik, M. New paradigms for computing, new paradigms for thinking. In *Computers and Explanatory Learning*; Springer-Verlag: Berlin/Heidelberg, Germany, 1995; pp. 31–43.
34. Ayodele, T.O. Types of machine learning algorithms. In *New Advances in Machine Learning*; IntechOpen: London, UK, 2010; pp. 19–48. ISBN 978-953-307-034-6.
35. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
36. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016; Volume 1, pp. 1–800, ISBN 978-0262035613.

37. Huang, G.B.; Wang, D.H.; Lan, Y. Extreme learning machines: A survey. *Int. J. Mach. Learn. Cybern.* **2011**, *2*, 107–122. [[CrossRef](#)]
38. Finocchiaro, M.A. Fallacies and the evaluation of reasoning. *Am. Philos. Q.* **1981**, *18*, 13–22.
39. Sigman, S.; Liu, X.F. A computational argumentation methodology for capturing and analyzing design rationale arising from multiple perspectives. *Inf. Softw. Technol.* **2003**, *45*, 113–122. [[CrossRef](#)]
40. Larvor, B. How to think about informal proofs. *Synthese* **2011**, *187*, 715–730. [[CrossRef](#)]
41. Marfori, M.A. Informal proofs and mathematical rigour. *Stud. Log.* **2010**, *96*, 261–272. [[CrossRef](#)]
42. Håkansson, A.; Hartung, R.; Moradian, E. Reasoning strategies in smart cyber-physical systems. *Procedia Comput. Sci.* **2015**, *60*, 1575–1584. [[CrossRef](#)]
43. Johnson-Laird, P.N. Deductive reasoning. *Annu. Rev. Psychol.* **1999**, *50*, 109–135. [[CrossRef](#)] [[PubMed](#)]
44. Psillos, S. Ampliative reasoning: Induction or abduction. In Proceedings of the ECAI96 Workshop on Abductive and Inductive Reasoning, Budapest, Hungary, 12 August 1996; pp. 1–6.
45. Hintikka, J. What is abduction? The fundamental problem of contemporary epistemology. *Trans. Charles S. Peirce Soc.* **1998**, *34*, 503.
46. Peirce, C.S. *Philosophical writings of Peirce*; Buchler, J., Ed.; Courier Corporation: Dover, NY, USA, 1955.
47. Aliseda, A. *Abductive Reasoning: Logical Investigations into Discovery and Explanation*; Springer Science & Business Media: Dordrecht, The Netherlands, 2006; Volume 330.
48. Meheus, J.; Verhoeven, L.; van Dyck, M.; Provijn, D. Ampliative adaptive logics and the foundation of logic-based approaches to abduction. In *Logical and Computational Aspects of Model-Based Reasoning*; Springer: Dordrecht, The Netherlands, 2002; pp. 39–71.
49. Satoh, K.; Inoue, K.; Iwanuma, K.; Sakama, C. Speculative computation by abduction under incomplete communication environments. In Proceedings of the Fourth International Conference on Multi Agent Systems, Boston, MA, USA, 10–12 July 2000; pp. 263–270.
50. Batens, D. A universal logic approach to adaptive logics. *Log. Univers.* **2007**, *1*, 221–242. [[CrossRef](#)]
51. Gauderis, T. Modelling abduction in science by means of a modal adaptive logic. *Found. Sci.* **2013**, *18*, 611–624. [[CrossRef](#)]
52. Menzies, T.J. An overview of abduction as a general framework for knowledge-based systems. In Proceedings of the Australian AI 1995 Conference, Canberra, Australia, 13–17 November 1995; pp. 1–8.
53. Hermann, M.; Pichler, R. Counting complexity of propositional abduction. *J. Comput. Syst. Sci.* **2010**, *76*, 634–649. [[CrossRef](#)]
54. Kean, A.C. A Formal Characterization of a Domain Independent Abductive Reasoning System. Ph.D. Dissertation, University of British Columbia, Vancouver, Canada, 1993; pp. 1–175.
55. Eiter, T.; Gottlob, G.; Leone, N. Abduction from logic programs: Semantics and complexity. *Theor. Comput. Sci.* **1997**, *189*, 129–177. [[CrossRef](#)]
56. Gottlob, G.; Pichler, R.; Wei, F. Bounded treewidth as a key to tractability of knowledge representation and reasoning. *Artif. Intell.* **2010**, *174*, 105–132. [[CrossRef](#)]
57. Poole, D.; Rowen, G.M. What is an optimal diagnosis? In Proceedings of the Sixth Conference on Uncertainty in AI, Cambridge, MA, USA, 27–29 July 1990; pp. 46–53.
58. De Campos, L.M.; Gámez, J.A.; Moral, S. Partial abductive inference in Bayesian belief networks using a genetic algorithm. *Pattern Recognit. Lett.* **1999**, *20*, 1211–1217. [[CrossRef](#)]
59. Psillos, S. Abduction: Between conceptual richness and computational complexity. In *Abduction and Induction*; Springer: Dordrecht, The Netherlands, 2000; pp. 59–74.
60. Pagnucco, M.; Foo, N. Inverting resolution with conceptual graphs. In Proceedings of the 13th International Conference on Conceptual Structures, Kassel, Germany, 18–22 July 1993; Springer: Berlin/Heidelberg, Germany, 1993; pp. 238–253.
61. Boutilier, C.; Becher, V. Abduction as belief revision. *Artif. Intell.* **1995**, *77*, 43–94. [[CrossRef](#)]
62. Kakas, A.; van Nuffelen, B.; Denecker, M. A-system: Problem solving through abduction. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Seattle, WA, USA, 4–10 August 2001; Volume 1, pp. 591–596.
63. Verdoolaege, S.; Denecker, M.; van Eynde, F. Abductive reasoning with temporal information. *arXiv* **2000**, arXiv:cs/0011035.
64. Sangiovanni-Vincentelli, A. Quo vadis, SLD? Reasoning about the trends and challenges of system level design. *Proc. IEEE* **2007**, *95*, 467–506. [[CrossRef](#)]

65. Gabbay, D.M.; Woods, J. Formal approaches to practical reasoning: A survey. In *Handbook of the Logic of Argument and Inference: The Turn Towards the Practical*; Elsevier: Amsterdam, The Netherlands, 2002; Volume 1, pp. 445–478.
66. Oaksford, M.; Chater, N. Conditional probability and the cognitive science of conditional reasoning. *Mind Lang.* **2003**, *18*, 359–379. [[CrossRef](#)]
67. Van Nuffelen, B. A-system: Problem solving through abduction. In Proceedings of the 13th Dutch-Belgian Artificial Intelligence Conference, Amsterdam, The Netherlands, 25–26 October 2001; Volume 1, pp. 591–596.
68. Kakas, A.; Riguzzi, F. Abductive concept learning. *New Gener. Comput.* **2000**, *18*, 243–294. [[CrossRef](#)]
69. Bylander, T.; Allemang, D.; Tanner, M.C.; Josephson, J.R. The computational complexity of abduction. *Artif. Intell.* **1991**, *49*, 25–60. [[CrossRef](#)]
70. Pourtalebi, S. System-Level Feature-Based Modeling of Cyber-Physical Systems. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2017.
71. Ruiz-Arenas, S. Exploring the Role of System Operation Modes in Failure Analysis in the Context of First Generation Cyber-Physical Systems. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2018.
72. Li, Y. Modelling, Inferring and Reasoning Dynamic Context Information in Informing Cyber Physical Systems. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2018.
73. Li, C. Cyber-Physical Solution for an Engagement Enhancing Rehabilitation System. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2016.
74. Horváth, I. Procedural abduction as enabler of smart operation of cyber-physical systems: Theoretical foundations. In Proceedings of the 23rd ICE/IEEE International Conference on Engineering, Technology, and Innovation, Madeira Island, Portugal, 27–29 June 2017; pp. 124–132.
75. Horváth, I.; Tepjit, S.; Rusák, Z. Compositional engineering frameworks for development of smart cyber-physical systems: A critical survey of the current state of progression. In Proceedings of the ASME Computers and Information in Engineering Conference, Quebec City, QC, Canada, 26–29 August 2018; pp. 1–14.
76. Tavčar, J.; Horváth, I. A Review of the Principles of Designing Smart Cyber-Physical Systems for Run-Time Adaptation: Learned Lessons and Open Issues. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 145–158. [[CrossRef](#)]
77. Pourtalebi, S.; Horváth, I. Information schema constructs for defining warehouse databases of genotypes and phenotypes of system manifestation features. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 862–884. [[CrossRef](#)]
78. Pourtalebi, S.; Horváth, I. Information schema constructs for instantiation and composition of system manifestation features. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 1396–1415. [[CrossRef](#)]
79. Ruiz-Arenas, S.; Rusák, Z.; Colina, S.R.; Mejia-Gutierrez, R.; Horváth, I. Testbed for validating failure diagnosis and preventive maintenance methods by a low-end cyber-physical system. In Proceedings of the 11th Tools and Methods of Competitive Engineering Symposium, Aix-en-Provence, France, 9–13 May 2016; Volume 1, pp. 1–11.
80. Ruiz-Arenas, S.; Rusák, Z.; Horváth, I.; Mejia-Gutierrez, R. Systematic exploration of signal-based indicators for failure diagnosis in the context of cyber-physical systems. *J. Front. Inf. Technol. Electron. Eng.* **2018**, 1–16, accepted.
81. Horváth, I.; Li, Y.; Rusák, Z.; van der Vegte, W.F.; Zhang, G. Dynamic computation of time-varying spatial contexts. *J. Comput. Inf. Sci. Eng.* **2017**, *17*, 011007-1-12. [[CrossRef](#)]
82. Li, Y.; Horváth, I.; Rusák, Z. Building awareness in dynamic context for the second-generation cyber-physical systems. *Future Gener. Comput. Syst.* **2018**, 1–34, submitted.
83. Li, C.; Rusák, Z.; Horváth, I.; Ji, L. Development of engagement evaluation method and learning mechanism in an engagement enhancing rehabilitation system. *Eng. Appl. Artif. Intell.* **2016**, *51*, 182–190. [[CrossRef](#)]
84. Li, C.; Rusák, Z.; Horváth, I.; Kooijman, A.; Ji, L. Implementation and validation of engagement monitoring in an engagement enhancing rehabilitation system. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *25*, 726–738. [[CrossRef](#)] [[PubMed](#)]
85. Wolpert, D.; Wheeler, K.; Tumer, K. *Collective Intelligence for Control of Distributed Dynamical Systems*; Technical Report NASA-ARC-IC-99-44; NASA: Ames, IA, USA, 1999; pp. 1–8.
86. Pease, A.; Aberdein, A. Five theories of reasoning: Interconnections and applications to mathematics. *Logic Log. Philos.* **2011**, *20*, 7–57. [[CrossRef](#)]

87. Tripakis, S. Compositionality in the science of system design. *Proc. IEEE* **2016**, *104*, 960–972. [[CrossRef](#)]
88. Fouquet, F.; Morin, B.; Fleurey, F.; Barais, O.; Plouzeau, N.; Jezequel, J.M. A dynamic component model for cyber physical systems. In Proceedings of the 15th ACM SIGSOFT Symposium on Component Based Software Engineering, Bertinoro, Italy, 25–28 June 2012; pp. 135–144.



© 2018 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).