Procedural abduction as enabler of smart operation of cyber-physical systems

Theoretical foundations

Horvath, Imre

DOI
10.1109/ICE.2017.8279879

Publication date
2017

Document Version
Final published version

Published in
2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)

Citation (APA)
Horvath, I. (2017). Procedural abduction as enabler of smart operation of cyber-physical systems: Theoretical foundations. In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (pp. 124-132). IEEE. https://doi.org/10.1109/ICE.2017.8279879

Procedural abduction as enabler of smart operation of cyber-physical systems:

Theoretical foundations

Imre Horváth
Cyber-Physical Systems Design
Industrial Design Engineering, Delft University of Technology
Delft, ZH, the Netherlands
i.horvath@tudelft.nl

***Abstract* — Some sort of 'intellectual' mechanisms are needed for second generation cyber-physical systems (CPSs) to be self-aware and self-adaptive. This paper (i) proposes procedural abduction (PA) as one possible mechanism for enabling smart reasoning, (ii) introduces the fundamental concepts and the computational procedure, and (iii) argues for their practical utility. PA includes eight cluster of activities: (i) run-time extraction of signals and data by sensing, (ii) recognition of events, (iii) inferring about existing situations, (iv) building awareness about the performance of the system at attaining of objectives of operation/servicing, (v) devising alternative performance enhancement strategies, (vi) designing adaptation of the parts and the system as a whole, (vii) planning the implied interventions, and (viii) actuating effectors and controls. As a computational approach, PA facilitates believes-driven contemplation of the momentary performance of a system with regards to the most relevant objective of servicing and 'best option'-based realization of adaptiveness. Computational realization of PA necessitates a combination of a large number of conventional and specific artificial intelligence algorithms. A fully fledged implementation of PA is underway, which will make verification and validation in the context of various smart CPSs possible.**

Keywords — second generation cyber-physical systems; smart operation; procedural abduction; contemplation and alteration; computational feasibility

I. INTRODUCTION

A. Background of the work

The paradigm of *cyber-physical systems* (CPSs) is rapidly developing and proliferating [25]. This is supported by: (i) the growing number of philosophical and theoretical studies [20], as well as by: (ii) the continuing striving for novel functionalities and services [22], (iii) the variety of enabling technologies and methodologies [4], and (iv) the increasing amount of systems implemented for practical applications [1]. The 'classical' definitions typically emphasized that CPSs are physical, chemical, biological and engineered systems whose operations are coordinated, controlled and monitored by a digital computing, communication and control core. However the latest definitions tend to emphasize the facts that CPSs: (i) employ the principles of *cyber-physical computing*, (ii) closely interact with the hosting environments, (iii) deeply penetrate into physical, biological, social, cognitive, etc. processes, and (iv) act as smart and adaptable actors in these contexts. CPC is deemed to overwrite the von Neumann theory-based computing (i.e. making calculations based on predefined data and algorithms). It works with run-time aggregated data, devises action scenarios according to changing objectives, and develops processing mechanisms based on recurrent cycles of sensing, reasoning, planning and actuating [15].

In one of our previous papers, we introduced the concept of *system generations* to impose an ontological framework on, and to differentiate and cluster the various implementations of CPSs [16]. For the sake of convenience of the reader, we mention here that identification of the generations of CPSs was based on two aspects: (i) the level of intelligence (relative to human intelligence, as ultimo maxima) and (ii) the level of self-organization (in terms of non-organic complexification). Based on these, five generations of CPSs were identified. The zeroth generation (0G) includes all forerunning ancestor systems such as advanced robotic systems and embedded (software) systems. The first generation (1G) comprises self-regulatory and self-tuning systems that can control/maintain their planned operation and achieve self-resilience in complicated situations. The second generation (2G) embraces CPSs that are able to realize some level of self-awareness and perform functional self-adaptation under varying conditions. The third generation (3G) is featured by self-cognizance (i.e. awareness with semantic understanding) and the capability of non-biological self-evolution. Regarded as the ultimate realization of CPSs, the fourth generation (4G) is supposed to have human-like system intelligence, which lends itself to self-consciousness-based understanding, fully dependable reasoning and decisional autonomy, and non-genetic self-reproduction.

The place of the various generations of CPSs on the landscape of systems is shown in Fig. 1. As mentioned above, awareness and adaptation are seen as the most important manifestations of smart operation of 2G-CPSs. However, the implementation of these capabilities necessitates some sort of system level '*intellectual mechanisms*'. Though there have

978-1-5386-0774-9/17/$31.00 ©2017 IEEE

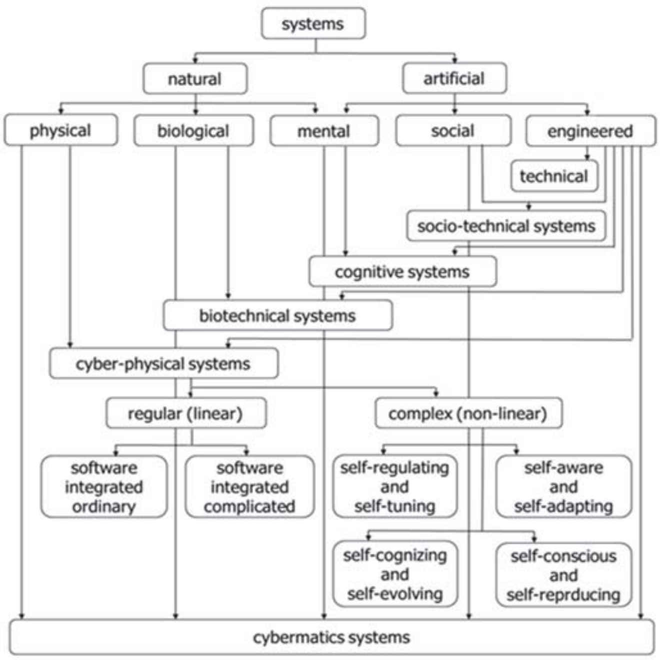

Fig. 1. The place of the CPSs generations on the landscape of systems

been efforts made in this direction in the field of artificial (general) intelligence research and development, the results are fragmented, non-transferable, and not interoperating from the perspective of 2G-CPSs.

B. Objectives of the presented work

Our background research concentrates on *cognitive engineering* of second generation of CPSs. This includes studying the principles of hybrid (model-based and run-time acquired) data-based system control, and developing mechanisms for (i) dynamic context management and system self-awareness, (ii) situated reasoning and strategy/decision making, (iii) not pre-planned self-configuration and self-adaption, and (iv) exploitation of 2G-CPSs in various application areas. Actually, we have been studying which forms self-developed awareness and adaptation can manifest in, and how they can be handled computationally in a synergetic and application independent manner. Our primary research hypothesis is that *procedural abduction* (PA) can be an integral computational mechanism for building awareness, development of operation strategies, making run time decisions, and actualization of adaptation. Beyond understanding the principles of contemplation and alteration by PA, we intend to develop a conceptual framework, a computational methodology, and a testbed implementation. In our conceptualization, PA is performed by an application independent, but flexibly instantiable contemplation and alteration mechanism (CAM), which can serve as a cognitive kernel for smart 2G-CPSs.

Related to this work, there are various preliminaries in the literature. They spread over four domains: (i) principles of artificial general intelligence and exploiting systems, (ii) context-sensitive formal reasoning mechanisms for smart systems, (iii) computational learning approaches for advanced industrial systems, and (iv) design and engineering of self-adaptive software. Smartness is claimed to be conditional for any form of (quasi)autonomous operation. In an oversimplified manner, Wikipedia explains that "smart systems incorporate functions of sensing, actuation, and control in order to describe and analyze a situation, and make decisions based on the available data in a predictive or adaptive manner, thereby performing smart actions".

System smartness is a relevant research topic for both *cognitive informatics*, which studies the internal information processing mechanisms and processes of the human brain and natural intelligence [11], and *cognitive computing*, which deals with smart mechanisms mimicking the structures and operational principles of the human brain and implements computational intelligence by autonomous inferences, intelligent computing methodologies, and alterable system configurations [24]. Human brain is typically explained and modelled by a hierarchical reductive structure, which extends to neurological (neuro-informatics), physiological (brain informatics), functional (cognitive models) and logical (abstract intelligence) levels. Development of smart systems raises many important questions in terms of computational implementation (knowledge processing, reasoning mechanisms, decision verification, human involvement, etc.). A typical current approach of equipping industrial systems with smartness is based on the use of multiple collaborative agents or networked problem solving agents.

C. Abduction in the related literature

Our research hypothesis has been introduced in Sub-section 1.B. and it will be shown in Section 2 that smart operation of a family of 2G-CPSs reflects the pattern of *abductive reasoning*. Proposition-based (logical) abduction has been widely studied [18] [3] [21]. Both problem-driven artificial intelligence (AI) research and brain reproduction-driven artificial general intelligence (AGI) research explored and exploited it in various noteworthy systems and application cases [7]. Nevertheless, many researchers argued that the exact nature of abduction and the principles for demarcating good and bad reasoning elements are still somewhat of a mystery. Prior literature is largely concerned with modeling how humans achieve abductive learning in the domain of formal logic or probabilistic logic [2]. Abductive reasoning received much attention from the point of view of formal reasoning, but much less as a possible reasoning mechanism for building awareness and operationalization of adaptation by 2G-CPSs themselves [6].

Conventional approaches to abductive reasoning are such as non-monotonic reasoning, circumscription, and default reasoning. These do not offer computationally processible ranking criteria with respect to the assumptions [10]. Typically, cardinality of the redundancy and relevance of the set-cover are considered as principles. Peng, Y. and Reggia, J.A. extended it with the goal of parsimony, to minimize the complexity of explanation [27]. Poole, D. claimed that abductive reasoning should be viewed as theory formation where logic tells us the consequences of our assumptions [28]. As a form of logical inference, abductive reasoning goes from an observation to a theory, which accounts for the observation. However, the observations (i.e. the collected elements of intelligence) do not guarantee a proper conclusion. That is the reason why abductive reasoning is regarded to as “inference to the best explanation”. Abduction is ampliative inference since the conclusion does not necessarily follow from the premises, but the conclusion amplifies the premises [14].

As approaches of technical implementation of abduction, (i) the logic-based abduction (LBA) approach and (ii) the set cover-based abduction (SCA) approach are the most often addressed ones in the literature. LBA implements reasoning through the use of a formal language and an appropriate logic over this language. The process starts out from a set of well-defined formulae and a set of observations. The objective of reasoning is to determine a particular set of hypothesis, which allows deriving the observations based on the domain theory and the underlying logic [17]. The logical truth or falseness of the observation is used to make decision on the conclusion [12]. SCA creates explicit (formally represented) sets of effects (E_i) and causes (C_j), and the interconnections (I_k) between them, that account for all of the effects observed. A cover is a set of those causes whose are related to the set of effects. The causal network formed by the interrelated observed manifestations, causal associations, and possible reasons (C_j-I_k-E_i) is the formal structure that underpins the reasoning, having the goal of covering all observed manifestations by proper explanations. In the case of causality-based reasoning the cause-effect relations described in a system operation model are used for semantic reasoning. This is however only a semantic difference between the two approaches, but not a difference of logic.

D. Content and structure of the rest of the paper

The rest of the paper provides a theoretical account on procedural abduction. Section II casts light on the roots of PA in the generic operation of a 2G-CPS, presents a formalization of the system operations, and provides formal definitions for the foundational concepts underpinning procedural abduction. Section III discusses the operation/servicing objectives of the system, presents both the generic activity workflow and the specific computational workflow of procedural abduction, and sketches up the sources of data and knowledge for a computational management of procedural abduction.

II. EMPIRICAL ROOTS AND FORMAL REPRESENTATION

A. Roots of PA in the generic operation of a 2G-CPS

Let us suppose that operation/servicing of a 2G-CPS is based on an overall learning-understanding-adaptation mechanism that can be generalized. Consider an application independent generalization of its operation/servicing as below. Let the 2G-CPS be equipped with a combined wired or wireless sensor network that provides analog signals and digital data. The signals are filtered and converted to digital data, based on which information about the status of achieving the functional objectives and the actual state of operation/servicing of the 2G-CPS can be generated. The set objectives and the planned operation are specified in an initial system operation/servicing model (SOM). Since the system constantly monitors the actual state of the real world processes it is concerned with, as well as its own architectural and operational states, any deviation from the system model-induced operation can be captured. It is probable that there are many concurrent changes and deviations in such a system of reasonable complexity. The 2G-CPS considers them together, interprets their interactions and impacts, and makes efforts to become aware of the emerged situation. In the process of interpretation it considers both the implicit contexts (objective) and the explicit context (circumstances and conditions) of operation/servicing. Having monitored the situation changes over time and in the shifting contexts, the 2G-CPS learns the tendencies of operation and achieving the goals of servicing.

Based on this knowledge and other (complementing) external knowledge (e.g. that is available in related ontologies), the 2G-CPS analyzes the interplay of the observed tendencies and tries to build up a reasonable understanding. Based on this it figures out what changes in the operation/servicing are needed in order to attain the preset objectives or even to set more promising ones. By projecting it to the actual operation/servicing, it determines if adaptation is needed and identifies what adaptation problem it does face (what and how to adapt?). In order to solve this problem, it develops multiple change plans. The various change plans are ranked according to certain principles such as: (i) the extent of adaptation they entail, (ii) the amount of resources needed to regain or enhance the objective of operation/servicing, or (iii) the opportunity of stating a better objective by a possible adaptation.

In its reasoning about the best change option, the 2G-CPS utilizes the bodies of information carried partly by its initial system model and partly by the updated system model actualized based on run-time acquired data, the initial context specifications and the run-time process dynamic contexts, and supporting knowledge repositories and ontologies. Having selected the best change plan, the 2G-CPS operationalizes the adaptation of the system as a whole. It determines what components (hardware, software and cyberware) are to be included, changed and/or excluded. It also determines what parameter settings are needed for the realization of the adaptation. Then, the 2G-CPS identifies all concerned effectors and sends the actuation information to them. This way, it introduces changes in the physical system, and ultimately in the real life processes. The execution of the designed adaptation and the altered system operation/servicing is monitored by the sensor network - thus the effects of adaptation are captured as soon as they become effective.

B. Formalization of the system operations

A specific reasoning pattern is hiding in the above-described overall system operation. This pattern is formed by the following elements of operation:

- Collect evidential data, facts, information, and knowledge about the overall operation and the context of operation of the 2G-CPS,
- Observe the fulfilment of the initial operational objective and the possibility of an improved objective and state of operation of the 2G-CPS,
- Develop alternative plans of operation, which consider the

objectives of operation and the implication of the collected information, and strive for the best possible operation of the 2G-CPS,

- Select the optimal plan of operation, which provides a better operational objective and system states for the 2G-CPS than any other one of the generated alternative plans of operation,
- Operationalize the optimal plan of operation for adaptation of the 2G-CPS assuming that it is probably correct in the given situation and context,
- Monitor if the introduced adaptation provides better results for the operation of the 2G-CPS by a continuing execution of the above steps.

While the first and the last steps are for aggregating elements of intelligence, the intermediate steps resemble the reasoning pattern of abduction that is well known from logic. However, since there are multiple definitions for logical abduction, we capture this pattern by the below definition: Let Σ_i be a set of observations (where $1 \leq i \leq N$), Q_j a set of explanations (where $1 \leq j \leq M$), and T a set of logically consistent interrelated theories representing a domain of knowing. With these:

Definition 0: *Logical abduction* (LA)
The logical process of deriving a set of explanations, Q_j, for a set of observations, Σ_i, according to a background theory, T, and picking out one best fitting, Q_b, of those explanations is called abduction. Δ

To be an explanation of Σ_i according to T, Q_i should satisfy two conditions, namely that: (i) Σ_i follows from Q_j and T, and Qj is consistent with T. 'The best fitting' explanation is selected based on criteria such as relevance, simplicity, probability, and assumed explanatory power.

C. *Foundational concepts for procedural abduction*

The cognitive framework of PA is based on a number of *foundational concepts* (FCs), which are actually interrelated by the framework. In our conceptualization, the FCs of PA are: (i) information, (ii) event, (iii) situation, (iv) context, (v) knowledge, (vi) awareness, (vii) learning, (viii) reasoning, (ix) strategizing, (x) stratagem, (xi) adaptation, and (xii) intervening. The FCs have been formally defined, from a computational perspective, based on knowledge from multiple literature sources, as follows:

***Definition 1**:* *Information* (I)
Information is the (cognitive) meaning determined by abstract artifacts (entities) and their associations (relations) that can be modeled, stored, and processed by human brains as well as by computing architectures. Δ

This definition reflects and is in line with the interpretation of information by cognitive informatics.

***Definition 2**:* *Event* (Ê)
Event is a happening of a change in signal flows and/or data streams of some importance during a particular interval of time, at a certain place, in a system operation mode, and in a given context. Δ

***Definition 3**:* *Situation* (Ŝ)
A situation is a particular combination, interrelationship and interaction of finite number of identifiable events as determined by the system's objectives as set-points, the internal states of affairs, and the external operational circumstances. Δ

***Definition 4**:* *Context* (Ĉ)
Context is a finite set of structure-able information associated with a part of reality (including things, relations, happenings, situations and circumstances) that have influence on the interpretation, meaning or effect of the rest of the reality. Δ

***Definition 5**:* *Awareness* (A)
Awareness is the state of (i) having knowledge concerning the existence and trends of situational changes, and (ii) formally 'understanding' their essence and implications. Δ

***Definition 6**:* *(System) knowledge* (K)
(System) knowledge is an arrangement of the total amount of *a priori* provided and *a posterior* information constructs acquired and applied to remove ignorance and/or uncertainty associated with existence and operation in the real world. Δ

***Definition 7**:* *Learning* (Λ)
Learning is the artificial ability of a software- and knowledge-enabled system to build not (explicitly) pre-programmed awareness and to acquire non-possessed information and/or knowledge under various control regimes. Δ

***Definition 8**:* *Reasoning* (Γ)
As computational version of human thinking, reasoning is the concurrently analytic and synthetic process of generating and using reasons (facts, evidences, or premises) to form conclusions, judgments, and/or inferences. Δ

***Definition 9**:* *Strategizing* (S)
Directing the operations/servicing of a system based on consideration of adaptation opportunities in order to maintain the achievement of a set of objectives, or setting and working according to a new, more relevant objective under changing conditions. Δ

***Definition 10**:* *Stratagem* (Ĝ)
An objective, context/situation dependent, elaborate tactical means/scheme devised to attain an actual objective. Δ

The stratagem of PA has two constituents: (i) the system adapts itself as a whole in order to optimize its operation, architecture, resources, and state towards the best performance, and (ii) the system selects its operational strategy by abductive reasoning.

***Definition 11**:* *Adaptation* (Â)
Any contemplated feasible alteration in the functionality, architecture, operation, implementation, interaction and servicing of a 2G-CPS as a whole, or any parts thereof that make the system better fitted to its predefined and/or emerging objectives in context. Δ

***Definition 12**:* *Intervention* (Î)
Intervention is the act of computing and setting the actually proper parameter information for the controllers of the physical and software effectors and converting it to control signals and data. Δ

Considering the above FCs from an epistemological point of view, we defined the whole process of procedural abduction as follows:

***Definition A**:* *Procedural abduction* (PA)

Procedural abduction is a computational process consisting of a *contemplation sub-process*, targeting acquisition of self-awareness related to a problem and a context, and an *alteration sub-process*, targeting the operationalization of self-adaptation of a 2G-CPS. Δ

Definition B: *Contemplation* (Ɔ)

Contemplation is a train of computational actions for (i) signal/data sensing/detecting, (ii) identification of events, (iii) interpretation of situations, and (iv) building awareness. Δ

Definition C: *Alteration* (Ɽ)

Alteration is a train of computational actions for (i) devising/selecting strategies, (ii) operationalization of adaptation, (iii) defining the interventions, and (iv) actuating effectors of the concerned 2G-CPS. Δ

PA facilitates smart operation and decisional performance of the embedding CPS. Smart operation concurrently means an objective- and context-orientated optimization of: (i) the state of managing the application objective of the system, and (ii) the state of achieving the relative most exploited operation of the system. These are mutually dependent on each other. Notwithstanding this fact, the state of managing the application objective of the system is the primary concern of the workflow. The needed proper state of operation of the system is ‘only’ a concomitant condition of an optimum realization of the objective. All together, these are described and specified by the *initial system operation/servicing model* (OSM_{initial}), developed in the design-time by the system designers with attention to the architecture, operation and control of the concerned system. As a complement, based on the skeleton of the OSM_{initial}, a *run-time adjusted system operation/servicing model*, OSM_{dynamic}, is also maintained, which captures the dynamics with regards to the changes in the objectives, the system states, and the environmental changes. The system’s OSM_{initial} and the changing operation state data, and the context data implied OSM_{dynamic} are the information technological basis of the implementation of PA. These were defined as below:

***Definition D**:* *Initial operation/servicing model* (OSM_{initial})

An initial operation/servicing model is a complete and prescriptive architecture, operation and service provisioning model of a 2G-CPS targeting a particular objective O_{initial}. Δ

***Definition E**:* *Run-time operation/servicing model* (OSM_{dynamic})

Generated based on the skeleton of the OSM_{initial}, an adjusted operation/servicing model is a data-induced dynamic up-date of OSM_{initial} targeting a particular objective O_{possible}. Δ

III. TOWARDS A COMPUTATIONAL IMPLEMENTATION

A. Consideration of operational objectives of the system

The intended (requested) operation/servicing performance is expressed by the *initial system objective*, $O_{\text{initial,}}$ determined by the designers of the system in the design-time. This cannot be taken as permanent, since it will change under dynamic conditions. The change (or in other words, the deviation from the target operation) is an observation that the system makes while operates. With regards to PA, observation concerns either the *emergence of a deviation* ($\Delta O_{\text{initial}}$) with respect to the initial objective O_{initial}, or the *emergence of an opportunity* for setting an enhanced (a more promising) objective ($\Delta O_{\text{possible}}$) relative to the initial objective O_{initial}. If the $\Delta O_{\text{initial}}$ remains below a ceiling value, then the system tries to come back to O_{initial}. However, when either $\Delta O_{\text{initial}}$ exceeds this value, or there is a chance to achieve a better overall system performance by considering a $\Delta O_{\text{possible}}$, the system sets a possible new objective, O_{possible}, by generating a new strategy for operation. This is made possible by the awareness having built up. The explanation on the situation is followed by generating a set of relevant *adaptation strategies* ($\hat{G}_k$) by the abductive reasoning mechanism of the system. The best explanation is an optimal-believed adaptation strategy chosen from this set of possible strategies, as the best stratagem. Formally:

***Definition F**:* *Best stratagem* ($\hat{G}_{\text{optimal}}$)

The best stratagem is one of the possible strategies, which attains the best performance in system operation/servicing with the smallest scale of adaptation and by a minimum resource usage. Δ

While working towards its objectives (O_{initial} or O_{possible}), a 2G-CPS may occasionally be confronted by observations inconsistent with its OSM_{initial} and/or OSM_{dynamic}. The CAM mechanism of PA is supposed to attempt explaining these observations by hypothesizing reasons for the inconsistencies and to use these in the development of the set of $\hat{G}_k$. Towards this end, it can rely on the awareness (inferred operational situation) obtained in the prevailing context. To be valid, the generated set of $\hat{G}_k$ needs to fulfil three conditions, namely:

(i) $OSM_{\text{initial}} \cup OSM_{\text{dynamic}} \cup \hat{G}_k \vDash O_{\text{initial}}$,
(ii) $\hat{G} \cup (PC \cup DC) \vDash O_{\text{initial}}$, and
(iii) $OSM_{\text{initial}} \cup OSM_{\text{dynamic}} \cup \Delta O_{\text{possible}}$ is consistent.

Evidently, other criteria can also be stated. The main characteristic of generating $\hat{G}_k$ is indefiniteness and this necessitates a syntactically and semantically convergent computational learning. It may well be that the knowledge incorporated in OSM_{initial} and OSM_{dynamic} is not sufficient to explain $\Delta O_{\text{initial}}$ and $\Delta O_{\text{possible}}$, or $\hat{G}_k$ and $\hat{G}_{\text{optimal}}$. In this case the process can be extended with forecasting (forerunning simulation) of the effects of a candidate $\hat{G}$. This knowledge does not enhance the power of the knowledge that is already included in OSM_{initial} and OSM_{dynamic}, but it makes the contemplation of the fulfillment of the objectives and the planning/realization of the adaptation more reliable and efficient. Obviously, generation of $\hat{G}_k$ also requires acquisition of conceptual knowledge. The concepts for the strategy are not discovered by a search for them, but by noticing discrepancies between the predictions OSM_{initial} and the implications of the run-time obtained and processed data, based on which the OSM_{dynamic} is derived. Thus, generation of $\hat{G}$ is a real time reasoning and computation process, (Γ).

To acquire the necessary awareness of the dynamics of the system objective and the changes of the context of operation (situation and state), the reasoning mechanism, (Γ), must deeply penetrate into the real life application processes, and elicit and aggregate process data. It should also be informed about the momentarily states of operation and servicing of the 2G-CPS. Interpretation of $\Delta O_{initial}$ and $\Delta O_{dynamic}$, the development of $\hat{G}$, and the choosing of $\hat{G}_{optimal}$ need the consideration of dynamic context information. Based on these, it has to build up the necessary awareness run-time.

B. The generic workflow of procedural abduction

According to Definitions A, B and C, the GWF consists of two sub-processes, which focus on attaining self-awareness and operationalization of adaptation, respectively. As sub-process of building awareness, the *contemplation activities* can be specified symbolically by a quadruple so as:

Ɔ = (DD, RE, IS, AA),

where: DD is detecting signals and elicitation of data by sensing material, energy and information flows, (I), in the concerned real-life processes/environment; RE is detecting, recognizing and monitoring events of operation, ($\hat{E}$); IS is interpretation of situations, ($\hat{S}$), caused by concurrent appearance and interaction of events in a particular dynamic context, ($\hat{C}$); and AA is learning tendencies (Λ) and acquisition of (decisional) awareness, (A), in the above context, ($\hat{C}$). As sub-process of operationalization of self-adaptation, the *alteration activities* are captured by a quadruple so as:

Ɽ = (DS, DA, PI, AE),

where: DS is devising possible operation/servicing strategies, ($\hat{G}$), and selecting the best adaptation strategy, ($\hat{G}_{optimal}$); DA is designing a combined architecture and operation adaptation plan, ($\hat{A}$), based on $\hat{G}_{optimal}$; PI is planning the interventions, ($\hat{I}$), with a view to all concerned effectors and system parameters; and AE is actuating the physical and computational effectors of the concerned CPS by providing the necessary control information. With these, the GWF of a 2G CPS can be described by the following ordered and oriented octuple (eight-tuple):

GWF = (DD » RE » IS » AA » DS » DA » PI » AE)

where: the symbols { DD, … , AE } are used in the same meaning as earlier, and the repeatedly used symbol '»' indicates the orientation of the flow of information processing. The overall generic workflow (GWF) of PA is graphically shown in Fig, 2.

The foundational concepts of PA are interrelated not only logically, but also computationally. The FCs imply high-level *computational activities*, each of which in turn includes varying number of low-level processing actions called *information transformations* or *transformations*, for short. The generic workflow of PA procedurally arranges the identified computational activities. On the other hand, the *transformations* included in the computational activities are arranged according to the internal logic of a particular activity. The transformations receive data directly from the transformations of other close-neighbor activities, and indirectly from interrelated far-neighbor activities. These informational interrelationships have been specified from procedural, methodological and information processing points of views, as reflected by the formal specifications in the next two subsections. Below, the definitions of the elements of PA will be further detailed from a computational implementation viewpoint.

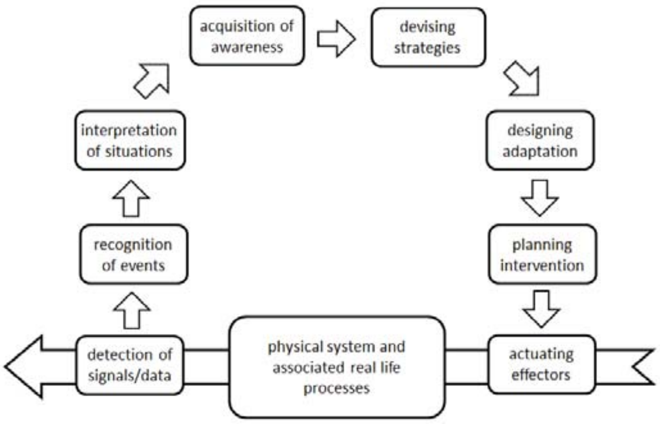

Fig. 2. Graphical representation of the generic workflow

C. The specific workflow of procedural abduction

OSM represents a 2G-CPS as a purposeful arrangement of a finite non-empty set of architectural entities and operation units on various level of aggregation and is defined as a sextuple:

OSM = (IV, AC, AR, OU, OR, OV)

where: IV is a finite non-empty set of input variables concerning architectural components and/or operation units, and their aggregations; AC is a finite non-empty set of architectural entities (components) on various level of aggregation; AR is a finite non-empty set of architectural relations among system components on the same or different levels of aggregation; OU is a finite non-empty set of operations/services provided by a non-empty subset of functionally interrelated architectural components; OR is a finite non-empty set of functional relations among components on the same and on various level of aggregation; and OV is a finite non-empty set of output variables concerning architectural components and/or operation units, and their aggregations.

The previous sub-section introduced the main activities that are necessary for the computational realization of PA. The system components and the architectural relations, as well as the functional elements and the functional relations are to be considered at defining the computational actions for both contemplation and alteration. Below we revisit each of these to provide more information on their computational implementations. We provide formal specification of each of the main activities and interpret the essence of the related information processing actions. In addition, further details will be provided concerning the use of operation/servicing model.

Each depicted activity of the GWF, including those belonging to its decision making sub-process, may be composed of two kinds of computational actions: (i) physically-grounded changes in the physical realm featuring continuous space and time, and (ii) data, information and knowledge processing activities in the cyber realm featuring digital representation and event-oriented execution. Both are seen as causes of state changes and are handled similarly as *transformations*, $T_{x,i}$, where: x is any activity of PA, and i is the identifier of a particular computational action belonging to x. It has to be noted that basic, auxiliary and interfacing data, information and knowledge (DIK) transformations can be identified from an information engineering perspective and that the above sextuple includes only basic DIK transformations. Auxiliary DIK transformations are such as recording digital data in files and visualization of data for human stakeholders, while interfacing transformations are such as manual human data input, or data conversion between cooperating system modules. The latter two kinds of transformation are not dealt with here due to space limitation.

Using the above formalism, the first activity, *detecting signals and elicitation of data* (DD), is defined as:

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

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 attributes of energy flows; $T_{DD,4}$ is local sensing 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.

The next activity involves the analysis of DIK to detect something that happens or might happen at a given physical or logical place and time related to the set objective of operation/servicing, the state of the components and that of the *system as a whole* (SaaW). Activities (ii) and (iii) focus on interpreting the situation with regards to the system's operations and states based on the interaction of events. The *recognizing and monitoring events* (RE) activity of the process is defined as:

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

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 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 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, *interpretation of situations* (IS) is defined as:

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

where: $T_{IS,1}$ is recalling all recognized events; $T_{IS,2}$ investigation of the space of the individual events in a considered local world based on the location of the signal provider; $T_{IS,3}$ 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 the identified situation.

Having a grasp on the status of objective achievement and system operation, the system intends to 'understand' the meaning and implication of this situation. This is based on learning (Λ), which allows the knowledge-enabled system to acquire additional information and to build awareness, and reasoning (Γ), which in turn allows making logical judgments and inferring conclusions. There are three sources of information: (i) the initial system model, (ii) the run-time acquired data, and (iii) the dynamic context model. Thus, the *acquisition of awareness* (AA) activity is computationally defined as:

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

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 AA and DA, this activity provides a logically thoughtful and coherent realization of machine-intuitive (non-fully algorithmic) thinking, which also informs about how and why the system came up exactly with a particular set of conclusions. 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 for the achieving the set objective is needed. This is the aim of the *devising strategies* (DS) activity, which includes the following computational transformations:

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

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 of 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 stratagem.

While strategizing focusses on 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 in produces a technical blueprint of the system alteration together with a course plan that is the basis of the intervention specification. The *designing adaptation*, (DA), activity is defined as:

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

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 $OSM_{dynamic}$ 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 $OSM_{dynamic}$ 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.

The DA activity results in a delayed adaptation due to the necessary preliminary testing of the influences of adaptation, and will output a sequence of adaptation actions. The objective of the *planning interventions* (PI) activity is to operationalize the refined adaptation plan. Actually it converts the adaptation blueprint into a *transition blueprint*, which considers the operations/servicing of the systems and the conditions for this. The PI activity is defined as follows:

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

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 *actuating effectors* activity (AE) is defined as:

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

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.

D. Some aspects of data and knowledge management

While the above GWF reflects the generic scheme of abductive logic as given in Definition 1, PA is actually not purely logical proposition-based, but information- and knowledge-based ('I+K-based') at once. This comes from the fact that CPSs implements a knowledge-intensive operation, typically based on formalized (and structured) system knowledge, (K). The overall scheme of using the knowledge sources is shown in Fig. 3. The major sources of knowledge for a practical execution of PA are: (i) the data, information and knowledge available and obtainable, respectively, concerning the objectives and operations of a particular 2G- CPS, (ii) the contents available in the $OSM_{initial}$, (iii) the contents generated in run-time for the $OSM_{dynamic}$, and (iv) the descriptive data and prescriptive constraints representing the permanent context (PC) and the dynamic context ($D\hat{C}$), and (v) specifications of conceptualizations in the associated system ontologies. It has been shown in our research that these bodies/chunks of knowledge are necessary, but also sufficient.

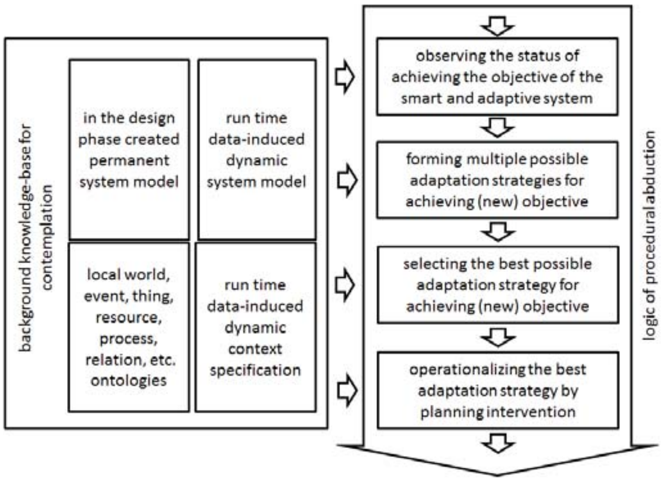

Fig. 3. The knowledge assets of procedural abduction

IV. CONCLUDING REMARKS

This paper does not claim to successfully implement a novel biology-based model of artificial general intelligence, but do claim to have made some important first steps in creating a framework for an ampliative reasoning mechanism for second generation cyber-physical systems. Our research is work-in-progress. The results summarized in this paper are related to the first phase of research, which concentrated on exploring the elements for a feasible conceptual framework and synthesized them from a computational perspective. We propose an approach based on which PA can be forwarded to a computational implementation and to functional, usability and utility testing. The specific aim of this paper is to describe the theoretical (notional) and computational fundamentals of a possible implementation of PA in the context of 2G-CPSs as a computationally feasible problem solving process. The proposed conceptual framework can be used in creating a procedural abduction kernel, which can be included in various hosting 2G-CPSs. This kernel will make them able to make decisions runtime and to adapt towards an optimal operation/servicing.

We have started the development of a computational methodology, the selection of ready-made IA tools, and the implementation of the needed new software tools, respectively.

The computational process of procedural abduction includes activities for awareness building, context-based decision-making, and adaptive behavior, in addition to a strategy-generation and -selection process. The pattern of logical abduction is explicitly recognizable in this activity, while the self-adjusting and self-optimizing behavior of CPSs is implemented though the whole of its contemplation-alteration operation. A major dilemma for artificial intelligence research is whether we want to include all cognitive capacities into systems in the design phase, or do we want to equip systems, with capabilities based on which they can develop their own intelligence. In the case of the 2G-CPSs the second strategy is proliferating. In line with the related efforts, our basic intension was to enable systems that may become more intelligent by learning from experience and that may adapt themselves in order to achieve the set or new goals. These systems should emulate and combine several human cognitive faculties to reach a set of non-completely defined (or even emergent) objectives. At the same time, the proposed reasoning mechanism differs from both the artificial general intelligence and the autonomous mental development projects.

REFERENCES

[1] Abbod, M.F., Linkens, D.A., Mahfouf, M. and Dounias, G. (2002). Survey on the use of smart and adaptive engineering systems in medicine. *Artificial Intelligence in Medicine*, 26(3), 179-209.

[2] Abe, A. (2008). Relationship between human and computational abduction. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, 12-15 October 2008, Singapore, IEEE, 318-323.

[3] Aliseda, A. (2006) Abduction as computation, in *Abductive Reasoning*, Springer Netherlands, 95-132.

[4] Andreou, A.G., Dykman, A.A., Fischl, K.D., Garreau, G., Mendat, D.R., Orchard, G., ... & Jackson, B.L. (2016). Real-time sensory information processing using the TrueNorth neurosynaptic system. In *Proceedings of the International Symposium on Circuits and Systems*, IEEE, 2911-2911.

[5] Babiceanu, R.F. and Seker, R. (2016). Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook, *Computers in Industry*, 81(9), 128-137.

[6] Bylander, T., Allemang, D., Tanner, M.C. and Josephson, J.R. (1991). The computational complexity of abduction. *Artificial Intelligence*, 49(1), 25-60.

[7] Codognet, P. (1997). Abductive reasoning: Backward and forward. In *Proceedings of the IJCAI'97 Workshop on Abduction and Induction in AI*, 13-16.

[8] Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., ... & Lastra, J.L. (2014). Industrial cloud-based cyber-physical systems. The IMC-AESOP Approach. Springer, 2013, 1-x.

[9] Donoso, M., Collins, A.G., & Koechlin, E., (2014), Foundations of human reasoning in the prefrontal cortex. *Science*, 344(6191), 1481-1486.

[10] Eiter, T. and Gottlob, G. (1995). The complexity of logic-based abduction. *Journal of the ACM*, 42(1), 3-42.

[11] Erlhagen, W., Mukovskiy, A. and Bicho, E. (2006). A dynamic model for action understanding and goal-directed imitation. *Brain Research*, 1083(1), 174-188.

[12] Eshghi, K. (1988). Abductive planning with the event calculus. In *Proceedings of the Fifth International Conference and Symposium*, Seattle, WA, August 15-19, 1988, MIT Press, Boston, x-x.

[13] Farid A. and Ribeiro, L. (2015). An axiomatic design of a multiagent reconfigurable mechatronic system architecture. *IEEE Transactions on Industrial Informatics*, 11(5), 1142–1155.

[14] Finin, T. and Morris, G. (1989). Abductive reasoning in multiple faults diagnosis. *Artificial Intelligence Review*, 3, 129-158.

[15] Hahanov, V., Hussein, M.A., Hahanova, A., & Man, K.L. (2016). Cyber physical computing. In *Proceedings of the East-West Design & Test Symposium*, IEEE, 1-8.

[16] Horváth, I., Rusák, Z., & Li, Y. (2017). Order beyond chaos: Introducing the notion of generation to characterize the continuously evolving implementations of cyber-physical systems. in *Proceedings of the ASME 2017 International Design Engineering Technical Conferences*, August 6-9, 2017, Cleveland, OH, IDETC2017/67082, 1-14.

[17] Jackson, P. (1989). Propositional abductive logic. In *Proceedings of the Seventh Conference on AISB*, 89-94.

[18] Jackson, P. (1990). Abduction and cuonterfactuals. In *Working Notes: AAAI Spring Symposium on Automated Abduction*. Stanford University, 77-81

[19] KIMERA Systems: (2016). Artificial general intelligence and the theory of intelligence. *White paper*, November 2016, Kimera Ssystems, Inc., Portland, OR, pp. 1-23.

[20] Knight, J., Xiang, J., & Sullivan, K. (2016). A rigorous definition of cyber-physical systems. In *Trustworthy Cyber-Physical Systems Engineering*, Chapter 3, CRC Press, 47-74.

[21] Konolige, K. (1990). Towards a general theory of abduction. In *Working Notes: AAAI Spring Symposium on Automated Abduction*. Stanford University. 62-66.

[22] Leitão, P., Colombo, A.W., & Karnouskos, S. (2016). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, 81(9), 11-25.

[23] Macías-Escrivá, F.D., Haber, R., del Toro, R. and Hernandez, V. (2013). Self-adaptive systems: A survey of current approaches, research challenges and applications. *Expert Systems with Applications*, 40(18), 7267-7279.

[24] Modha, D.S., Ananthanarayanan, R., Esser, S.K., Ndirango, A., Sherbondy, A.J., & Singh, R. (2011). Cognitive computing. *Communications of the ACM*, 54(8), 62-71.

[25] Oks, S.J., Fritzsche, A., & Möslein, K.M. (2017). An application map for industrial cyber-physical systems. In *Industrial Internet of Things*, Springer International Publishing, 21-46.

[26] Paul, G. (1993). Approaches to abductive reasoning: An overview. *Artificial Intelligence Review*, 7(2), 109-152.

[27] Peng, Y. and Reggia, J.A. (1990). Abductive inference models for diagnostic problem-solving. Springer-Verlag, Berlin.

[28] Poole, D. (1989). Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5, 97-110.

[29] Reniers, M., van de Mortel-Fronczak, J., & Roelofs, K. (2017). Model-based engineering of supervisory controllers for cyber-physical systems. In *Industrial Internet of Things*, Springer International Publishing, 111-136.

[30] Vogel, T., & Giese, H. (2014). Model-driven engineering of self-adaptive software with EUREMA. *ACM Transactions Autonomous and Adaptive Systems*, 8(4), 18:1-18:33.

[31] Vogel, T., Seibel, A., & Giese, H. (2011). The role of models and megamodels at runtime. In *Models in Software Engineering*, pp. 224-238.

[32] Wang, Y. (2003). On cognitive informatics. *Brain and Mind*, 4(2), 151-167.

[33] Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., & Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291(5504), 599-600.