First steps towards a mereo-operandi theory for a system feature-based architecting of cyber-physical systems

Shahab Pourtalebi, Imre Horváth, Eliab Z. Opiyo

Faculty of Industrial Design Engineering
Delft University of Technology
Landbergstraat 15
2628 CE Delft, the Netherlands
s.pourtalebihendehkhaleh@TUDelft.nl
{i.horvath;e.z.opiyo}@TUDelft.nl

Abstract: Cyber-physical systems (CPSs) are characterized by a high-level synergy among hardware, software and cyberware constituents. Though there has been intense research in various fields of complex engineering systems, still no advanced and comprehensive system design methodologies are available that would facilitate design and development of CPSs in various contexts and applications. This paper reports on the first steps and results of developing a comprehensive trans-disciplinary theory for conceptualization, modeling and realization of system architectures through a unified epistemological platform. The essence of the proposed theory is the fusion of mereotopological, modus operandi and CPS knowledge.

1 Design challenges of cyber-physical systems

As a manifestation of emerging highly complex systems, cyber-physical systems (CPSs) are rapidly proliferating. CPSs tightly integrate hardware, software and knowledgeware, are deeply embedded in the application environment, and typically, they are characterized by: (i) structural heterogeneity and complexity, (ii) operational synergy of components, (iii) nonlinear behavior, (iv) decentralized real-time operation capabilities, and (v) open architecture and ad-hoc functional connections between components. Although these characteristics provide completely new capabilities for CPSs, they also imply that the traditional methods and principles cannot realistically be utilized in design and development of CPSs [PHO14].

Our work concentrates on cyber-physical consumer durables (CPCDs). As a specific sub-category of CPSs, CPCDs are product-service combinations that reflect most of the functional and structural features of cyber-physical systems, while enabling a higher level of interaction with users and the embedding environments. The research reported here focuses on exploring the possibilities of designing customizable CPCDs and enabling this by some sort of methodological and/or computational means. Our previous investigation found that achieving the goals of mass customization (MC) in the case of

CPCDs is rather challenging [PHO13]. Additionally, it has been argued that due to their new system features, we need new design principles for MC of CPCDs.

The challenges of customization of CPCDs appear in combination with the challenges emerge due to the differences of approaches that are used for hardware, software and cyberware constituents, and due to the variety of aspects in which the heterogeneity should be addressed (i.e., control, sensing, actuating, exploration, informing, networking, adaptation, safety, security and reliability).

2 Why do we need a new methodology for architecting CPCDs?

The currently available approaches and methodologies of complex system development are typically mono-disciplinary, limited in terms of the scope of application, and do not guarantee compositionality of a CPCD as a whole. Nevertheless, components-based and platform-based architecting is becoming the conceptual and procedural standard for conceptualization, design and development of CPSs. The advantage of this approach is that it can be applied equally well to hardware, software and cyberware constituents. Considering these facts our intention is to develop a trans-disciplinary, widely applicable and system synergy-oriented approach for MC of CPCDs.

Our assumption has been that based on a common epistemological platform (i.e., underpinning theory) we may be able to bring the three domains in overlap and to develop a methodology that does not differentiate them. Obviously, the sought-after underpinning theory can only be based on new concepts. Having recognized the importance of this, we decided to extend the well-known concept of form and application features to the level of the constituents of the complex heterogeneous system.

In the system architecture context, product features are features of sub-systems and components. While, system features are features of the whole system. Product features can be application features or form features (Fig 1). Every component of a physical product is manifested in a certain form. These features are morphological definition of a component which can be distinguished by their material, surface quality, and everything that is related to shape of a part. The application features, by contrast, define realizations and applications of a component, i.e., design features, concept features, and manufacturing features. These definitions can be extended to software products as well. In this sense, codes, algorithms and programming language can be resembled to form features, and software development process features and software design features can be considered as application features.

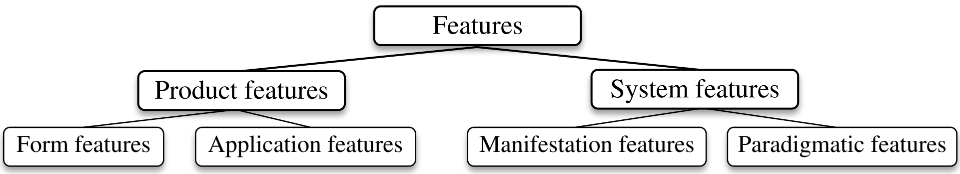

Figure 1: different kinds of features

System features are the result of components features as a composition. A system feature is a construct that include both paradigmatic features and manifestation features, which complement each other on different abstraction levels. We defined paradigmatic system feature (PSF) as a logically-based and physically-based abstraction of a system as whole that differentiates it from other comparable systems. These features are rooted in inherent characteristics of a system. A manifestation system feature (MSF) is related to a specific system in contrast to paradigmatic features that are about common features of a group of systems that share some similarities. They can be manipulated by designers and cannot violate overall system features.

Though system features are more related to the operational aspects of systems, product features are more associated by the structural architecture of components. Therefore, we need to address features composition by combining structural and operational architecture. In this sense, any existing artifact can be categorized in terms of these categories of features through operational and structural architecture.

3 Fundamentals of a new mereo-operandi theory

In order to develop an epistemological platform, it seemed to be necessary to blend various component theories that were able to capture the essential aspects of architecting CPCDs. In our research three originally distinct bodies of formal knowledge namely: (i) the theory of mereotolopology, (ii) the theory of engineering modus operandi, and (iii) the theory of CPSs have been combined in developing a unified theory.

3.1 Applied concept of mereotopology

Mereotopology was used to describe the part-whole relationships of operational domains of systems. Mereology literally means "science or theory of parts" [KYK08]. The name came from the formal theory of parts and associated concepts developed by Leśniewski [Ls82]. Topology, by contrast, implies ontological laws pertaining to the boundaries and interiors of wholes to relations of contact and connectedness, and so on [Sb96]. In the theory of mereotopology, mereology generally implies part-whole relation, while topology describes component connectedness. In our epistemological platform, mereotopology could play an important role in describing structural and operational architecture of CPSs. In this sense, relation of components-system and components together can be formalized and defined by mereotopological interpretations.

3.2 Applied concept of engineering modus operandi

By "engineering modus operandi" we refer to three things together: operation, functionality and behavior. If we consider operations as tangible outcomes of a system and the ways of delivering that output by sub-systems, we can describe its function at a higher level of generality and abstraction. In fact, operation explains how a function is accomplished. However, behavior defines the expected function of a system in its context. Then, we need to consider operations of a system and its components in order to understand how the desired function is delivered, and the system behaves in its context.

In considering the operation of a system, we need to understand the operational architecture. "Operational architecture" is defined as the procedural architecture of a domain for realization of an operation. The architecture depends on the morphological characteristics of constituents actively involved in that operation. Additionally, the operational architecture and the morphology of constituents are supported by natural phenomena (e.g. gravity, friction, etc.). So, we need to consider three other levels in order to explain the operation of a system or a domain (Fig 2).

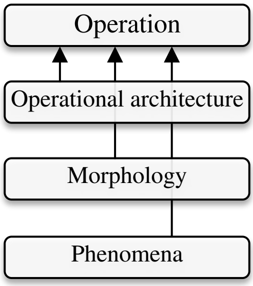

Figure 2: Elements of operation

3.4 Integrating the concepts

The theory of cyber-physical systems is used to take care of system compositionality and component composability issues. From the system compositionality view point, it can be noted that system output is the aim which could be realized by components. However, from the components composability viewpoint, emphasis is on the components which should work together in order to deliver the desired output of the system. Since both viewpoints can be followed in designing and developing CPCDs, compositionality and composability challenges should be considered for architecting of CPCDs. So, it is imperative to apply hierarchical decomposition of operational domains.

In order to describe manifestation system features of CPSs, a system can be represented based on several basic elements (as domains of operation) that define its operation. We named these basic elements as "operational building blocks" (OBBs). These OBBs should be (i) applicable for modeling of a large group of CPSs, (ii) able to be categorized operational-wise, (iii) based on simple rules for modeling, (iv) modular, (v) storable to form a repository, and (viii) open source (to be developed for different purpose). The operational and structural architecture of OBBs can be defined by mereotopological architecture when we are considering their relationship with their peers in the same level of granularity or their parental relationship, as well as engineering modus operandi when we want to explain procedural operation structure. We defined three main flows in domains of operation to consider operational processes, namely, material flow, energy flow and information flow. Material flow donates every kind of physical operation (physical input, output and manipulation of OBBs). Energy flow defines every kind of energy transmission, generation and consumption in OBBs. Information flow presents every kind of information processing, communication, sharing and storing in OBBs.

4 First results with validating the proposed mereo-operandi theory

A CPS as a system ($\sum$) consists of domains of operation (D_i), $\sum=\{D_1, D_2, \ldots, D_n\}$. Domains of operation and their sub-domains can be represented by OBBs. Although hierarchical relations can be represented (through nesting) by OBBs, the degree of granularity depends on the level that is required for modelling of a system. Features of an OBB can be determined by defining D={I,M,O} where I is the input of a specified domain, O is the output of that domain, and M is the manipulation which is done by the domain. In this sense, M(I)=O.

There are three kinds of input/output for representing material flow, information flow and energy flow. Different components of a system can be inferred by their inputs and outputs and/or combination of them. In light of this interpretation, for modeling a CPS, software, hardware or cyberware components can be modeled in a uniform manner, and based on the kind of flows they receive and send as input (I) and output (O).

A system and its components can be decomposed into OBBs. Then, we can introduce features of a system and its components by defining their OBBs' input, output and manipulation. Input and output of an OBB can be determined by its specifications (e.g. force (F), voltage (V), communication protocol (P), etc.). $I=\{ F_i, V_i, P_i,\ldots\}$, $O=\{ F_o, V_o, P_o,\ldots\}$. As well as manipulation of an OBB which could be defined by its specifications such as processing power (PP), produced noise (PN), energy consumption (EC), customization capability (CC), strength (S), durability (D), failure probability (FB), and any other aspects from designing, developing, customization, maintenance, and other perspectives. $M=\{PP_m, PN_m, EC_m, CC_m, S_m, D_m, FB_m, \ldots\}$

From composability viewpoint, manifestation features of a system can be defined by composing its OBBs' features. From compositionality viewpoint, manifestation features of a system can be decomposed for defining features of its OBBs. So, the proposed theory can satisfy both perspectives of design, development and architecting of CPSs. By modeling a system through OBBs, (i) specifications of each OBB in terms of input, output and manipulation can be formulated. These formulations could explain form feature and application feature mentioned in Section 2. (ii) Various operations of the system can be simulated based on different use scenarios. So, manifestation features of the system can be identified. (iii) Moreover, numerous open source algorithms can be developed and utilized in order to find opportunities for customizing the system, discovering probability of system failure, identifying bottlenecks and week points of the system, determining kind and level of user interaction and so on.

It is important to note that three libraries are needed to support our modeling approach. The first one is the product feature library. This library includes codes which define specifications of OBBs. It can be generated by producers of the components or other developers (i.e. translating product specification into codes). The second library is the manifestation system features library that could be provided by system designers or system architects. It shows various operation scenarios and considers system behavior in different context. The third library is the methodology and tool development library. This library contains design principles, development methods, maintenance principles, user-interaction regulations, etc., which could be developed by tool developers or knowledge engineers in form of some algorithms.

The major issue in our research is to show embedded customization capability in a system. It simply means having an OBB in our system with customizable M. This customizable M means using different kinds of manipulations which can result in multiple operations: $M=\{f_1, f_2, f_3, \ldots\}$ and $f_1(I)=O_1$, $f_2(I)=O_2$, $f_3(I)=O_3$, etc., and it is worth mentioning that different indexes of O might mean different channel of output, different values of same output or even using different physical output ports. Various manipulations of embedded customization might be stored in an open source library.

5 Conclusion

The main objective of our research was to develop a new theory, and methodology and/or tools for modeling CPSs. To achieve this objective, we needed to create an integrated epistemological foundation and a framework that support the development of a design methodology and a toolbox of enablers.

We have argued that, in addition to its mereotopological specification, MSFs are also characterized in terms of composability and compositionality, as well as of their operation. The physical operation is described in terms of the underpinning physical phenomena and principles, the morphology of the related domain, and the procedural structure of the operation. The OBBs can be used in modeling manifestation features, which carry mereo-operandi information, has specific meaning in particular operations, implementations and application contexts, and can be synthesized in alternative ways.

The developed theory, among others, can: (i) formally and symbolically present MSFs, (ii) represent hardware, software and cyberware constituents of CPSs in a uniform manner, (iii) allow specification and simulation of material, energy and information flow of CPSs, and support designing and implementation of CPSs. It is expected that the new theory will be able not only to integrate the currently disjointed hardware, software and cyberware knowledge, but will also bring the abstraction-based conceptualization and component/platform-based implementation of CPSs closer.

References

[PHO14] Pourtalebi, S.; Horváth, I.; Opiyo, E.: New features imply new principles? deriving design principles for mass customization of cyber-physical consumer durables, Proceedings of TMCE 2014, May 19-23, 2014, PP. 95-108, Budapest, Hungary.

[PHO13] Pourtalebi, S.; Horváth, I.; Opiyo, E.: Multi-aspect study of mass customization in the context of cyber-physical consumer durables, Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Portland, Oregon, USA.

[Hi98] Horváth, I.: Shifting paradigms of computer aided design, Delft University Press, Delft, The Netherlands, pp 1–36.

[KYK08] Kim, K.; Yang, H.; Kim, D.: Mereotopological assembly joint information representation for collaborative product design, Robotics and Computer-Integrated Manufacturing, Volume 24, Issue 6, December 2008, Pages 744-754.

[Ls82] Leśniewski, S.: On the foundations of mathematics, Topoi, 2 (1982), pp. 7–52.

[Sb96] Smith, B.: Mereotopology: A theory of parts and boundaries, Data & Knowledge Engineering, Volume 20, Issue 3, November 1996, Pages 287-303.