Engineering of Cyber-Physical Systems*

María Victoria Cengarle
fortiss GmbH
cengarle@fortiss.org

Abstract

The present work reflects on the drivers and barriers of Cyber-Physical Systems with focus on the challenges associated with their engineering. Based on a previous survey, the charachteristics are studied that a reference framework for the engineering of Cyber-Physical Systems should support. Thereby orthogonal dimensions of the envisioned systems and proposed system development phases, on the one hand, and diverse solutions in use, on the other, are collated.

1 Introduction

The present work refers insights and thoughts concerning the mastering of Cyber-Physical Systems (CPS). Nowadays CPS are touted as the next revolution in Computer Science. Forerunners can already be found in as dissimilar areas as automotive, avionics, energy, health, environmentalism and consumer electronics. The vision poses extraordinary challenges particularly regarding technology, organisation and human-system cooperation. It also entails a huge potential both for economy as well as for tackling problems of modern society.

Here we focus on the engineering challenges posed by CPS, and draft some raw ideas on how to meet those. We firstly introduce in Sect. 2 our definition of CPS. Secondly in Sect. 3, we perform a kind of meta-requirements analysis in order to find out the demands a reference framework for CPS must fulfil. Afterwards, in Sect. 4 we detail candidate methods for the different phases in which we divide the development process of CPS. Finally in Sect. 5 we draw some conclusions and outline a number of possible improvements of the currently available loose ends.

2 Cyber-Physical Systems

As defined in [4], a Cyber-Physical System (CPS)[1] is a system with embedded software (as part of devices, buildings, means of transport, transport routes, production systems, medical processes, logistic processes, coordination processes and management processes), which:

- directly records physical data using sensors and affect physical processes using actuators;
- evaluates and saves recorded data, and actively or reactively interacts both with the physical and digital world;
- is connected with other CPS and in global networks via digital communication facilities (wireless and/or wired, local and/or global);
- uses globally available data and services;
- has a series of dedicated, multimodal human-machine interfaces.

The result of the connection of embedded systems with global networks is a wealth of far-reaching solutions and applications for all areas of our everyday life. Subsequently, innovative business options and models are developed on the basis of platforms and company networks. Here, the integration of the special features of embedded systems –for example, real-time requirements– with the characteristics of the internet, such as the openness of the systems, represents a particular technical challenge.

The main objective of the project “Innovation Platform Cyber-Physical Systems Engineering” is the integration, validation, and dissemination of a coherent, ready-to-use reference framework based on state-of-the-art science and technology as well as on the design and operation life cycle continuum of CPS Engineering (CPSE). It is also intented to instantiate and validate the CPSE by means of cross-domain scenarios and case studies. The long-term

*This work has been partially sponsored by the EIT ICT Labs, project “Innovation Platform Cyber-Physical Systems Engineering”.

[1] The term CPS is here used both as a singular and a plural noun, the number depending on the context.

vision is to establish the EIT ICT Labs as the cross-domain, multidisciplinary open innovation platform for developing and maintaining a ready-to-use, open and standardised CPSE reference framework, that facilitates the transitioning of complex and trustworthy CPS to the marketplace.

Challenges and opportunities

The biggest challenge brought about by the engineering of CPS is the integration of (discrete as well as continuous) models and methods from different disciplines including not only technical ones like mechanical and electric/electronic engineering, computer science[2] and control theory but also ergonomics and human factors, economic ecosystems, social guidelines and legal stipulations. These "soft" aspects of CPS are crucial for the acceptance of CPS and therefore for their success.

In exchange, there are a number of very significant opportunities allowed for by CPS. Besides value creation and innovation, the most noticeable ones are the enhancement of accident prevention procedures, improved support of ageing population, and smart use of limited resources. These have been considerably emphasised in [4, 13].

Regarding only the computational discipline, on the one hand we have traditional Business Information Systems (BIS) and, on the other, traditional Embedded Systems (ES). The former are data-centric, ideally high secure and open, focus on maintenance, their life cycle incorporate legacy systems and evolves continuously, and their constraints fall in the category of weak real-time. Their engineering challenges are application integration, enhancement of running systems, re-engineering of legacy systems, and validation and prediction. The latter, on the contrary, are function-centric and closed, focus on construction, their life cycle consists of decommission followed by design, building and commission (i.e., legacy is not an issue), and their constraints fall in the category of hard real-time. Their engineering challenges are systems engineering (function, architecture, platform, and mechanics), safe function deployment, and verification; see [5] and also [9]. By CPS these both sorts of systems need be combined; considering their description above, it is redundant to stress that their reconciliation is anything but straightforward. Moreover, the large-scale dimension of CPS aggravates the situation.

Engineering discipline

Because of the considerations above, it becomes apparent that the Engineering of CPS (CPSE) calls for a radical new paradigm allowing the integrated construction, operation, adaptation and evolution of large-scale, long-living, heterogeneous, open, dependable (in particular, safe and secure), high-investment systems. There is a series of aspects to be considered for devising a new CPSE reference framework; see [5]. On the one hand, we have the continuous life cycle of CPS that amalgamates Integrated Development Environments (IDE)[3] and Operating Systems (OS)—and thus puts a combined functionality at disposal for the design, simulation and verification, deployment, operation, maintenance of CPS. In this context longevity, including self-documentation, self-reflection, self-adaptation and self-optimization, as well as criticality, i.e. uninterrupted operation modifiable at runtime, need be meaningfully provided for.

On the other hand, built-in support is necessary for dependability including safety and security ("BIS meets ES") as well as large-scale: built-in compositionality for construction and operation of CPS (thus confronting the larger-scale knot posed by CPS). Moreover, the envisioned CPS engineering must include online-models of system, environment and situation and of domain-views as well.

3 Demands on the engineering

Existing reference frameworks for, e.g., embedded systems and systems of systems do not address the new challenges posed by CPS: openness and heterogeneity, portability and interoperability across domain boundaries as well as situation awareness and self-evolvability, among others. However, some already existing networked embedded systems let the conjecture raise of the suitability of upgrading and aggiornamento of established embedded systems engineering frameworks.

Starting point for the development of a proposal is here an analysis of the demands to be observed by a suitable reference framework for the development of CPS. Such a framework has to reveal cross-cutting fundamental scientific and engineering principles that underpin the integration of cyber and physical elements across the addressed sectors. The tentative concepts of the CPSE reference

[2] Computer science here encompasses in many cases only unsatisfactorily solved issues as, e.g., interoperability, adaptability and tailoring, learning, private data protection, fault tolerance, safety and security.

[3] IDEs are applications that facilitate efficient software development by providing not only a syntax-oriented source code editor but also build automation (compilation, linking, deployment, etc.) and debugging tools.

framework include life-cycle processes, terminology, design principles, guidelines together with an adapted multiview framework.

Depending on the degree of tightness (cf. Fig. 1), cooperating systems can be viewed as different and cooperating CPS or as together composing a single (however big) CPS. The latter materialise in the case of diverse, even wide spread, but cooperating development teams; the former when, e.g., systems initially not meant to cooperate with each other are composed. The engineering issues vary accordingly. In the first case, we speak of "coarse-grained" aspects, while in the second we speak of "fine-grained" concerns.

3.1 On the coarse-grained level

Open and dynamic systems can be bundled in order to provide a service that may be realised by not a single but a chain of systems spontaneously cooperating. The thus arising systems' cooperation poses the implicit challenge of the identification of individual systems as well as the description of the service offered by these. What moreover means that an orthogonal modelling dimension is indispensable for dynamic and spontaneous cooperation.

As pointed out in [4, 13], the approach must rely, particularly during requirements analysis, on an interdisciplinary approach. CPS systems stem from most diverse domains and are operated by people about whose background almost no assumption can be made, thus ease of use of those combined systems is imperative. An orthogonal dimension of design, therefore, has to consider the presentation aspects of each of the systems as well as their tie points.[4]

The conjecture here is that these three design modelling activities, namely individual service(s) design, cooperation design and presentation design, be separated thus supporting modular and reusable design. Furthermore, the importance has been recognised of involving users in the innovation process, be they sources of innovative ideas, early testers in simulation environments, or even developers; see [2]. Living labs let researchers and engineers test and modify products in close collaboration with end-users in a real-life or a real-as-life setting. Living Labs capture users' insights, prototype and validate solutions, aim to contribute to both problems providing structure and governance to the user involvement and methodologies and organizations to filter and sense user insights; see [11]. Two prominent living labs are FutureEverything and the city of Oulu, Finland. FutureEverything is an art and digital innovation organization based in Manchester, England, around an annual festival of art, music and digital culture, that each year presents the work of around 300 artists across its art, music and conference strands, and is conceived as a living lab for participatory experiments on art, society and technology; see [15]. The world's first wristwatch rate monitor, GSM telephone call, WCDMA telephone call, etc. came from Oulu, whose dynamism is due to the Innovation and Marketing group of the city that acts as a Living Lab, setting up and analysing user experiences and laying out the service model; see [2]. Furthermore, in [21] the benefits are shown of children's participation in living labs.

3.2 On the fine-grained level

As already mentioned, it is unclear how conventional modelling techniques for BIS and for ES can be sensibly combined. One prominent problem profusely treated in the literature refers to modelling techniques addressing heterogeneity and hybridity (e.g., discrete vs. continuous models); see [10, 18, 24]. Also the integration of successful techniques for one realm into the other, for instance component-based engineering into ES design, as pointed out in [8] among others, is anything but obvious. In the microscopic level of embedded systems, dependability issues represent a non-trivial challenge that apparently cannot be enough warned of. This issue is magnified when one considers that, dynamic and spontaneously, services communicate and cooperate as an action or a reaction to the situation. Thus, mechanisms for authentication as well as for intention and need recognition ought to be improved/perfected or even devised where nonexistent.

4 CPS Reference Framework

Aligned with the methodologies that were successful so far, we conceive a development methodology for CPS divided into phases that are to be composed and combined iteratively and successively and taking into account the different levels of refinement of single units as well as the different degrees of maturity of interacting systems. The purpose of this section is twofold: On the one hand, the above mentioned phases are seen from the perspective of CPS and, on the other, some specific (or dedicated) pro-

[4] Cooperation and presentation are akin to the navigational design and the presentational design web and hypermedia applications described in [16].

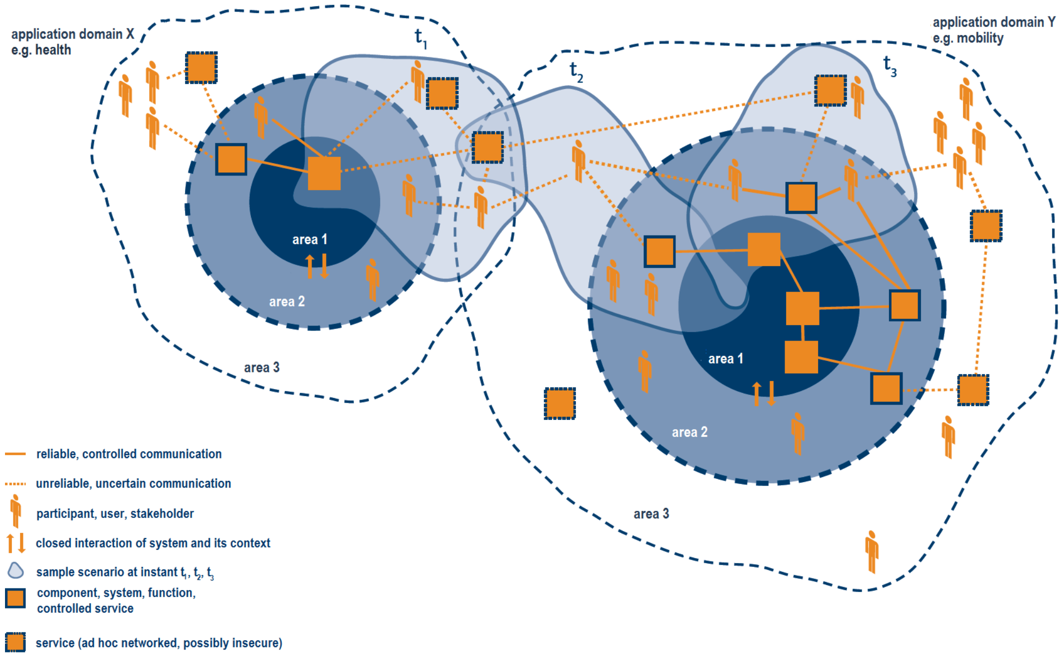
application domain X
e.g. health
t_1
t_2
t_3
application domain Y
e.g. mobility
area 1
area 2
area 3
area 1
area 2
area 3
reliable, controlled communication
unreliable, uncertain communication
participant, user, stakeholder
closed interaction of system and its context
sample scenario at instant t_1, t_2, t_3
component, system, function,
controlled service
service (ad hoc networked, possibly insecure)

Figure 1: CPS domain structure (Source: [13])

posals and solutions nowadays in use are suggested that could cope with the task at hand.

The initial stage of any system is customarily called requirements analysis. Already for this first approach to a system there is in the realm of CPS (at least) two fundamentally orthogonal approaches. These are discussed in Sect. 4.1 below.

Reflecting on the further phases that can constitute a sound and all-encompassing reference framework for the development of CPS, we recognise different ways of approaching the challenges depending on the point of view assumed. The core application of a system (of systems) can be termed *service*, whose nature appears to be one from three possibilities: business, computation and control, or platform; see Sect. 4.2. The interplay between systems (of systems) requires the definition of rules for cooperation also comprising authentication, service description and communication protocols; see Sect. 4.3. Crucial for the acceptance of these by far non-obvious systems is the way they communicate with end-users, their ease (i.e., intuitiveness) of use, and their possible customisation; see Sect. 4.4.

4.1 Requirements Analysis

Similar to web applications, also CPS "facilitate business process integration, new business models, supply chain mediation disintermediation and reengineering, as well as offer new services to new markets"; see [19]. And likewise "potential users are so diverse and geographically wide-spread that [requirements analysis strategies predicated on consulting the future users of the systems] is impractical." In the cited work, thus, an approach to requirements elicitation is proposed that "combines the recognition of multiple user views of a complex human activity system with techniques to help creatively map existing and potential business functions to a Web-based environment [. . .] accessible to developers who are not IT function experts." This method, termed SSM/ICDT, combines the Soft Systems Methodology (SSM, see [22]) with the ICDT matrix (information, communication, distribution and transaction, see [3]). Because of the similarities mentioned, valuable insights may be gained by considering an activity-oriented approach to requirements analysis of CPS.

Alternatively, artefact-based approaches "promise to provide guidance in the creation of consistent artefacts in volatile project environments, because these approaches concentrate on the artefacts and their dependencies, instead of prescribing processes"; see [20]. The conducted a case study which showed the increased flexibility of the approach in comparison with the previously used one, as well as the improved quality of the created artefacts, and also that productiveness was not improved. A so-called mega-modelling environment termed Global Model Management (GMM, see [27]) permits typing, composition and execution of artefacts. As a consequence, type errors during execution can be avoided. This approach, appropriately transferred to the CPS setting, could used for authentication prior the establishment of a spontaneous cooperation.

4.2 Service(s) Design

It turns out that at least three kinds of services converge into CPS, that interact with each other. They are depicted in Fig. 2. An hypothesis worth considering is that the above difficulties of combining BIS and ES be solved by decoupling systems and let them only communicate via a platform (i.e., removing the dashed arrows in Fig. 2). This very probably implies a platform with considerably more intelligence than that of conventional ones.

Much has been said and written about the immense costs of (fine) tuning, maintenance, and redesign and re-engineering of large scale complex IT systems. As stated in [23], "management structure that moves a megaproject along with seamless transitions between the project's phases can affect the final outcome and success".

Computation / control services

Because of its two dimensions of abstraction, the SPES Metamodel [14] seems an adequate starting point for the embedded systems dimension of service(s) modelling. On the one hand, there are the software development perspectives and, on the other, the levels of granularity; see Fig. 3. The former permit the examination of a system from different viewpoints and this way gain or specify diverse kinds of information about the system.

The functional perspective describes the systems from the angle of its usage. The logical perspective describes the system as a network of communicating and cooperating components possibly hierarchically structured. The technical perspective provides the technical details of the system especially with regards to hardware and virtual machine platform, and is conceived in such a way that it can be extended (or instantiated) for particular application domains.

The SPES Metamodel supports a process-based system development (see also [28]), into which the operation-design continuum of CPS may not fit smoothly. An alternative to be considered is a system development based, rather than on processes, on products; see [7].

Business services

Business services have been extensively addressed, at the beginning in manifold ways; see [1]. There are different approaches in this realm, so for instance Business Process Management (BPM), Service-Oriented Architecture (SOA), Service-Oriented Computing (SOC), etc.; see, e.g., [29].

Platform services

A platform is much more than just a vehicle of information. It probably has the job of managing huge amounts of information, without neglecting their integrity and confidentiality. It moreover has to mediate between systems of inherently diverse nature. Worth considering for the realisation of these services is the solution proposed by the middleware Chromosome; see [6] and also [17]. Chromosome returns the control over the functionality of an application to the developer, by "hiding" the complexity and ensuring extensibility by plug & play mechanisms also at runtime. Chromosome moreover puts real-time capabilities at disposal.

4.3 Cooperation Design

Cooperation between single systems and CPS (i.e., cooperation at any level, see Sect. 3) can be organised considering the concepts of navigation space and navigational structure; see [16]. This means, navigation nodes and navigability between nodes are notions orthogonal to component and communication between components. Navigation moreover can (but not necessarily does) carry information. An intuitive example is given by smartphones, where for instance email reading can be interrupted by an incoming call: once the call ended, the control automatically returns to the previous screen and the user can continue reading his/her email. Information navigates when, e.g., within the email an address is selected and the smartphone offers to look up the address using a map service, to search for a connection with public transportation, etc.

Service delegation can moreover occur dynamically. This means, the situation can be assessed and required services be searched for in the surroundings of the system. With regards to the concepts of navigation nodes and navigability, the structure can spontaneously be redefined and/or rearranged.

Typically, a navigation structure is based on the services structure. The former, however, can omit some services, i.e., not necessarily all services are reflected –or represented– as a navigation node; an

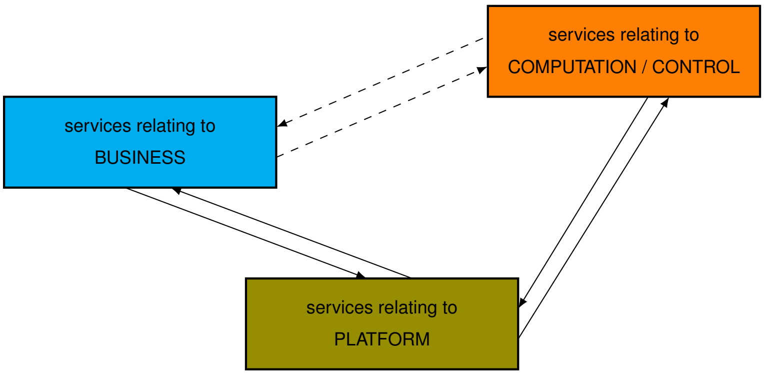

Figure 2: The three kinds of services

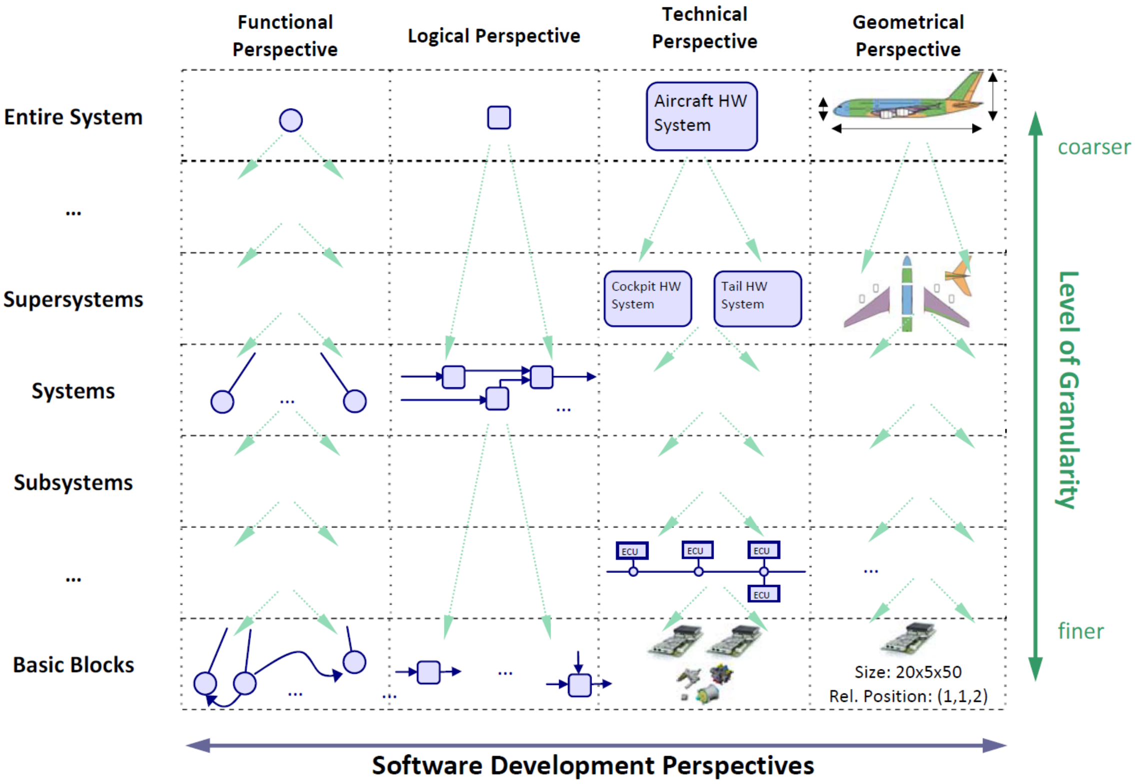

Figure 3: The two abstraction dimensions of the SPES Metamodel (Source [14, 26])

example hereof might be an antivirus. Navigation can also foresee shortcuts avoiding intermediate steps when these are, e.g., previously and univocally determined.

4.4 HCI Design

The success of web services lets us presume that, for the interaction of CPS with end users regardless of their education, age, gender, etc., the strategies used in web design can be reused. For this purpose, the models for presentation of [16] may be a good starting point; see also [12].

The presentation model is based on the navigational model, but addresses other challenges. Consider haptic in case of an amputee, or instructions imparted to hearing impaired, etc. These considerations greatly impact on the acceptability of CPS; see [4, 13].

5 Conclusions and outlook

The design-operation life-cycle continuum of CPSE reminds of a family album, where snapshots are memorised but in fact the portrayed subjects might exist beyond the ends of the album, i.e., some exist before the first (in chronological order) photograph was taken, some exist further after the date of the last photograph, and some others happen to appear as grown-ups when, e.g., a family member marries. Moreover, between two chronologically subsequent photographs, any family member has undergone a number of more or less slight changes.

Referring back to the SPES Metamodel, the first to catch one's eye is the compartmental division between software development perspectives: although the time unfolds from left to right, it is not to be understood that all levels of granularity of a CPS evolve simultaneously from one perspective to the next one. The picture misses moreover the correlation between the components across the different perspectives. On these realisations and considering the discussion above we plan to work out a process and a metamodel for CPS and to iteratively validate them by means of case studies.

References

[1] W. M. P. Aalst, A. H. M. Hofstede, and M. Weske. Business Process Management: A Survey. In W. M. P. Aalst and M. Weske, editors, *Business Process Management International Conference (BPM'03, Proceedings)*, volume 2678 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2003.

[2] E. Almirall and J. Wareham. Living Labs and open innovation: Roles and applicability. *The Electronic Journal for Virtual Organizations and Networks*, 10(3):21–46, 2008. Special Issue on Living Labs.

[3] A. Angehrn. Designing mature internet business strategies: The ICDT model. *European Management Journal*, 15(4):361–369, Aug. 1997.

[4] M. Broy, E. Geisberger, M. V. Cengarle, P. Keil, J. Niehaus, C. Thiel, and H.-J. Thönnißen-Fries. *Cyber-Physical Systems: Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion*. Number 8 in acatech BEZIEHT POSITION. Springer, Berlin, 2012.

[5] C. Buckl, H. Rueß, and B. Schätz. Cyber-Physical Systems: From Building to Evolving — A Radically New Engineering Challenge of European Dimension. Presentation at the EIT ICT KIC "Cyber-Physical Systems", Feb. 2012.

[6] CHROMOSOME in 120 Minutes. Technical report, fortiss GmbH, Apr. 2012.

[7] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional, 3rd edition, 2001.

[8] I. Crnkovic. Component-based approach for Embedded Systems. In *9th International Workshop on Component-Oriented Programming (WCOP'04, Proceedings)*, 2004.

[9] B. Curtis, M. Kellner, and J. Over. Process Modeling. *Commununications of the ACM*, 35(9):75–90, 1992.

[10] P. Derler, E. Lee, and A. Sangiovanni Vincentelli. Modeling Cyber-Physical Systems. *Proceedings of the IEEE*, 100(1):13–28, Jan. 2012.

[11] B. Dutilleul, F. Birrer, and W. Mensink. Unpacking European Living Labs: Analysing Innovation's Social Dimensions. *Central European Journal of Public Policy*, 4(1):60–85, 2010.

[12] F. Garzotto and V. Perrone. On the Acceptability of Conceptual Design Models for Web Applications. In M. Jeusfeld and O. Pastor, editors, *Conceptual Modeling for Novel Application Domains (ER'03 Workshops ECOMO, IWCMQ, AOIS, and XSDM, Proceedings)*, volume 2814 of *Lecture Notes in Computer Science*, pages 92–104. Springer, 2003.

[13] E. Geisberger, M. Broy, M. V. Cengarle, P. Keil, J. Niehaus, C. Thiel, and H.-J. Thönnißen-Fries. *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*. Springer, Berlin, 2012.

[14] A. Harhurin, F. Hölzl, and T. Kofler. SPES Metamodel. Deliverable D1.2.B-6, Software Plattform Embedded Systems (SPES) 2020, Dec. 2010.

[15] D. Hemment, R. Ellis, and B. Wynne. Participatory Mass Observation and Citizen Science. *Leonardo*, 44(1):61–63, 2011.

[16] R. Hennicker and N. Koch. A UML-based Methodology for Hypermedia Design. In A. Evans, S. Kent, and B. Selic, editors, *UML 2000 :The Unified Modeling Language, Advancing the Standard (3rd International Conference, Proceedings)*, volume 1939 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2000.

[17] K. Huang, G. Chen, N. Keddis, M. Geisinger, and C. Buckl. Demo Abstract: An Inverted Pendulum Demonstrator for Timed Model-Based Design of Embedded Systems. In *Third International Conference on Cyber-Physical Systems (ICCPS'12, Proceedings)*, page 224. IEEE Computer Society, 2012.

[18] G. Karsai, J. Sztipanovits, Á. Lédeczi, and T. Bapty. Model-Integrated Development of Embedded Software. *Proceedings of the IEEE*, 91(1):145–164, Jan. 2003.

[19] M. Meldrum and J. Rose. Activity Based generation of requirements for web-based information systems: the SSM/ICDT approach. In T. Leino, T. Saarinen, and S. Klein, editors, *The European IS Profession in the Global Networking Environment, 13th European Conference on Information Systems (ECIS'04, Proceedings)*, pages 1212–1223, 2004.

[20] D. Méndez Fernández, K. Lochmann, B. Penzenstadler, and S. Wagner. A case study on the application of an artefact-based requirements engineering approach. In *Evaluation Assessment in Software Engineering (EASE'11, Proceedings)*, pages 104–113. IEEE Computer Society, 2011.

[21] M. Reichel and H. Schelhowe. Living labs: driving innovation through civic involvement. In J. Cassell, editor, *7th International Conference on Interaction Design and Children (IDC'08, Proceedings)*, pages 141–144. ACM, 2008.

[22] J. Rose. Soft systems methodology as a social science research tool. *Systems Research and Behavioral Science*, 14(4):249–258, July/August 1997.

[23] T. Sorel. The Life Cycle Continuum. *Public Roads*, 68(1), July/August 2004.

[24] J. Sztipanovits. Model Integration and Cyber-Physical Systems: A Semantics Perspective. Invited talk at FM'2011, June 2011. Joint work with Ted Bapty and Gabor Karsai and Sandeep Neema.

[25] V. Teglasi. Why Transportation Mega-Projects (Often) Fail? Case Studies of Selected Transportation Mega-Projects in the New York City Metropolitan Area. Master's thesis, Columbia University, New York, USA, 2012.

[26] J. Thyssen, D. Ratiu, W. Schwitzer, A. Harhurin, M. Feilkas, and E. Thaden. A System for Seamless Abstraction Layers for Model-based Development of Embedded Software. In G. Engels, M. Luckey, A. Pretschner, and R. Reussner, editors, *Software Engineering Workshops 2010 (Proceedings)*, volume 160 of *Lecture Notes in Informatics*, pages 137–148. Gesellschaft für Informatik, 2010.

[27] A. Vignaga, F. Jouault, M. C. Bastarrica, and H. Brunelière. Typing artifacts in megamodeling. *Software & Systems Modeling*, 12(1):105–119, 2013.

[28] Y. Wang and A. Bryant. Process-Based Software Engineering: Building the Infrastructures. *Annals of Software Engineering*, 14(1-4):9–37, 2002.

[29] M. Wirsing and M. Hölzl, editors. *Rigorous Software Engineering for Service-Oriented Systems*, volume 6582 of *Lectuer Notes in Computer Science*. Springer, 2011.