TUM

TECHNISCHE UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK

Reference Framework for the Engineering of Cyber-Physical Systems: A First Approach

Denis Bytschkow, Alarico Campetelli, María Victoria Cengarle, Maximilian Irlbeck, Konstantin Schorp

TUM-I143

Reference Framework for the Engineering of Cyber-Physical Systems: A First Approach

Alarico Campetelli, Maximilian Irlbeck

Technische Universität München, Department of Informatics
Boltzmannstr. 3, 85748 Garching, Germany
{campetel, irlbeck}@in.tum.de

Denis Bytschkow, María Victoria Cengarle, Konstantin Schorp

fortiss GmbH
Guerickestr. 25, 80805 München, Germany
{bytschkow, cengarle, schorp}@fortiss.org

Abstract

The connection of Embedded Systems with global networks is a wealth of far-reaching solutions and everyday applications. The integration of the special features of Embedded Systems, like, e.g., real-time requirements, with the characteristics of the Internet, such as openness, poses a technical challenge. The main objective of the project CPSE is the integration of a coherent Reference Framework. Inspiration is found in different disciplines including not only technical ones like mechanical and electric/electronic engineering, computer science and control theory but also ergonomics and human factors, economic ecosystems, social guidelines and legal stipulations. These latter aspects of CPS are crucial for the acceptance of CPS and therefore for their success.

Keywords: Cyber-Physical System, Reference Framework, Reference Architecture, Embedded System, Open System, Smart Grid, Automotive

---

Contents

1 Introduction

The transition from embedded systems to Cyber-Physical Systems (CPS) was triggered by the networking of the former. The possibilities thus opened break the mould of ordinary embedded systems, and consequently the hitherto proposed engineering frameworks fail to address specific challenges of CPS as, e.g., their openness. This very characteristic means that CPS are portable, interoperable with each other, heterogeneous and self-evolvable, to name a few particulars. Self-evidently, this very much departs from the traditional application area of embedded systems. Moreover, the rigour displayed by the development of embedded systems that often are safety critical, cannot be relaxed to the licences frequently shown by the realization of Internet applications.

The present document investigates the matter in further detail, and proposes certain lines of actions in order to bridge the gap between known methods and one suitable for the realization of the envisioned systems. The ultimate goal is a reference framework for the engineering of CPS. This is beyond the possibilities at hand, but a discussion can certainly be let loose by the statement below. And, should be this the case, it would be very welcome, since the CPS vision can only be achieved if there is a wide understanding on how these systems can interoperate, what in turn means that a common or at least a concerted and coordinated approach to CPS interaction (and development as well, most likely) is necessary.

1.1 General Procedure

Our expertise allows us to generalize experiences from the Smart Grid and Automotive sectors and to propose a transfer of these to other CPS domains. More precisely, characteristics and empirical values common to those domains are extrapolated. In doing so, individual dimensions are described and the attention is drawn to the challenges that arose and most probable are to be found in any CPS domain as well. The considered dimensions are:

- Human socio-technical dimension: role of humans as part of as well as outside systems, limits of modelling, what to trust: human or machine; regarding this very last aspect, guidelines and regulations are imperative
- Technical realization of systems: cloud architecture, distribution of systems, management of large volumes of distributed data, high availability of services
- Technical realization of communication channels: channels (glass fiber, wireless, powerline), bandwidth, quality of service, performance, availability, security against eavesdropping, dedicated vs. public networks, property rights and liability

- Data Dimension: collection of data (as measured by sensors, among others), query frequency, push/pull mechanisms, data storage, aggregation, preparation and processing, data mining, durability, unlinkability[1], anonymization, substitute values, plausibility checking, precision, trigger mechanisms, data dependencies
- Security: access to data, authentication, authorization, non-repudiation, IT Security, information privacy
- Criticality dimension: redundancy, reliability, fault tolerance, black start capability
- Architecture: key functions, structure, data flow, interfaces, stakeholder-specific or application-specific views (complexity reduction, abstraction, distraction reduction, access protection), rationales, cross reference domains, reference architecture, functional, logical, and technical levels
- Physical World: sensors, actuators, repercussion on physical systems, sampling, spatial/temporal aspects
- Legal and social dimension: laws, regulations, by-laws and their impact on system design
- Market/business view: market structure, systems and market roles, business objectives, business processes
- Terminological view: denomination of system components, cross-domain communication, conceptualization

These dimensions were considered according to the own expertise, and the insights gained in doing so were taken into account for the reference framework presented in section 9.

1.2 Reference Frameworks

There are many software reference frameworks, development processes, philosophies, methodologies (or methods[2]), and best practices. Design patterns for

---

[1] Unlinkability refers to a data management requirement that calls for the separation of data and processes from different contexts. As far as possible, data should not be inferred at all. The aim is to prevent the risks connected with the accumulation of data that can be analysed in a variety of different ways and to restrict the purposes for which such data may be evaluated. Unlinkability also means that data from separate contexts should be kept separate and not processed as a single data chain. Other protection goals are intervenability and transparency. By intervenability it is understood that the persons concerned actually have the possibility to exercise their rights. A system is transparent to the persons concerned as well as the persons who run it if its functionality and its effect are intelligible to a sufficient grade and, moreover, provided those persons can retrieve the data available to the system with reasonable expenditure.

[2] According to [50], “In normal usage, a method is an approach to accomplishing a task, and a methodology is the study of a family of methods. Within the software community, the term methodology usually denotes an approach to accomplishing a task.”

instance constitute a well-known way of documenting solutions to design problems, where a pattern must explain why a particular situation causes problems and why the proposed solution is considered a good one. Relatively popular is the Unified Process, an iterative and incremental software development process that can be customised before use by any organization. The Rational Unified Process (RUP) is a refinement of the Unified Process created by Rational Software Corporation (now a division of IBM). RUP was simplified into the Agile Unified Process (AUP) which is based around four phases: inception, elaboration, construction, and transition. Agile Software Development, shortly very popular, tries to minimise risk by developing software in brief iterations that in themselves are small projects. There is also the Framework for Software Product Line Practice [14], a web-based document that makes use of patterns as a way of expressing common contexts and problem-solution pairs and can be used to show how aggregations of practice areas can be orchestrated to solve recurring problems.

Also for embedded systems there is a plethora of proposals. In [22], for instance, many frameworks are listed and a further development of Model-Driven Engineering (MDE, see [54]) for embedded systems is presented. And the list can be arbitrarily extended. To name a few, one finds the Enterprise Architecture framework, the Web application framework, the Component-Based Systems-of-Systems reference framework (CBSoS; see [39]), model-driven development of component-based distributed real-time and embedded systems (see [55]), model-based systems engineering (see [49]), etc. [3]

Frameworks promise higher productivity and shorter time-to-market (than non-framework-based approaches) through design and code reuse; see [51]. Although this referred work is concerned with object-oriented frameworks, it puts forward an interesting thesis on framework design, i.e., the focus is not only on the activities of the framework users but also on those of the expert framework developer(s). Therein a framework is defined as a model of a particular domain or an important aspect thereof. A framework may model any domain, be it a technical domain like distribution or garbage collection, or an application domain like banking or insurance. A framework provides a reusable design and reusable implementations to clients.

1.3 Scope of this Article

As mentioned above, the experiences gathered in two domains as diverse as Smart Grid and Automotive are carefully examined. Considering the challenges these domains pose, commonalities are identified as well as possible

---

[3] A seemingly quite exhaustive list of frameworks can be found at `http://www.realsoftwaredevelopment.com/the-complete-list-of-software-development-frameworks-processs-methods-or-philosophies/` (last access January 15th, 2014).

approaches for meeting them. The proposed solutions include interchange using protocols, functionality and interfaces, human factors, etc. As a result, requirements (at the meta-level) are recognized that have to be fulfilled by any method for the realization of CPS. Moreover, a Reference Architecture and a Reference Workflow as well as a checklist are exemplarily sketched. These take into account specific issues as, e.g., context and openness, legacy systems, and behavioural integrity assurance. The contoured Reference Framework, or at least its critical consideration and a reflection on its advantages, provides support for devising a common language (for experts' exchange) and for collaborative interaction, as well as the basis for a standard across national boundaries, that addresses not only technical aspects like dynamic and spontaneous collaboration but also social aspects as for instance acceptance.

What is beyond the scope of this work is a validation of the framework. For this purpose, more than just one case study, of moreover industrial scale, would be absolutely necessary. This departs from the extent of the document and of the project.

A Reference Framework for the engineering of CPS should reconcile traditional Business Information Systems, which are data-centric and open, focus on maintenance (legacy is an issue), and their constraints fall in the category of weak real-time, with traditional Embedded Systems, that are function-centric and closed, focus on construction (legacy is not an issue), and their constraints fall in the category of hard real-time.

2 CPS Description

The original definition of these emerging systems dates back approximately eight years, when a group of academics in the United States of America recognized the evolution of embedded systems whose physical aspects gained more and more importance. So much so, that the interaction between networked and distributed processors of growing complexity, on the one hand, and the physical world enwrapping them, could no longer be considered of minor relevance. The term Cyber-Physical System (CPS[4]) describing the research field has since then gone through a number of finer clarification attempts and seen a plethora of further sample applications joining the family of instances of CPS. As defined in [23], a CPS 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 above list of systems cannot be exhaustive: further application domains are manufacturing, entertainment, consumer appliances, chemical processes, and civil infrastructure, to name a few. Also the list of characteristics turns out to be incomplete: real-time capabilities, distributed and/or shared control[5], context adaptivity and (partial) autonomy, organization as system of systems could be mentioned as well. In addition, and addressing the "physical" part of the name, the notion of "directly" recording seems restrictive, a CPS needs only the means, be these direct or not, to retrieve and not necessarily to record sensed information, and to make use of one or more actuators in order to change the state of a controlled device. It can be immediately recognized that, further refining and adding to this definition will certainly

---

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

[5] In a shared control approach, multiple systems and humans co-operate in an effective way by adopting. Shared control is not to be confused with distributed control; see, e.g., [66].

result in an overly detailed list of features and applications.[6]

The understanding of CPS in the rest of this document is as follows:

> *A **Cyber-Physical System (CPS)** consists of computation, communication and control components tightly combined with physical processes of different nature (e.g., mechanical, electrical, and chemical), and understood (and evaluated) in a social and organisational context.*

2.1 Technology Push Perspective

Novel systems with integrated computational and physical capabilities, that moreover interact with humans through manifold modalities, enable a plethora of technology-based developments with significant impact on society. The possibility of (directly or indirectly, spontaneous or planned, statically or dynamically) interact with, and expand the capabilities of, the physical world through computation, communication, and control is a key driver for future technology improvements. Still embryonic progresses, that unclose opportunities as well as challenges, can be already observed in as diverse application domains as transportation, health and well-being, and energy, to name a few. So much so that Cyber-Physical Systems have been coined the next wave of innovation in information and communication technology.[7] Here we can include the design and development of next-generation airplanes and space vehicles, hybrid gas-electric vehicles, fully autonomous urban driving, and prostheses that allow brain signals to control physical objects.

2.2 Demand Pull Perspective

The demands of economy and society inexorably and disruptively lead to value-added chains and economic ecosystems spanning over diverse domains. Indeed, innovation in technology and the consequent technology push occurs in tight interaction with a corresponding demand pull. Products that use those technologies and are demand conforming, meet the acceptance of the market and end users. Indeed, technological innovations are not only supply-side driven. Rather, demand pull is extremely influential: the more intense the demand, the more creative groups and individuals are drawn to work on unsolved problems; see [33].

End users customarily use highly sophisticated devices and appliances, and are very demanding. Not surprisingly, thus, an increased ecological sensibility

---

[6] The buzzphrase is “self multi live”, meaning self-X (self-monitoring, self-healing, self-documenting, etc.), multi-functional, multi-domain, multi-technology, etc., as well as live (re)configuration, life update, life (re)deployment, etc.

[7] This new wave of innovations and changes could constitute (or be aligned with) the much cited and discussed sixth wave of innovation; see e.g. [34,40,57,59].

raised the claims for the use of renewable energy sources and the decentralisation of energy generation. This promoted investment in research and development of, e.g., solar panels for individual households as well as smart meters. Energy prosumers, i.e., end users that themselves generate energy,[8] may demand energy from the supplying company when their consumption exceeds their production, or may alternatively feed the energy grid with their energy surplus. That means, they can buy as well as sell energy. The energy meter needs likewise be upgraded, in such a way that both directions of energy flow are measured, and must also track other relevant information like date and time, etc.

2.3 Summary

The technology push anticipates demand, and can be termed "market seeding", whereas a demand pull provokes a reaction from the technology providers, whose reaction can be termed " market satisfaction". As stated in [47],

> The promise of CPS is *pushed* by several recent trends: the proliferation of low-cost and increased-capability sensors of increasingly smaller form factor; the availability of low-cost, low-power, high-capacity, small form-factor computing devices; the wireless communication revolution; abundant internet bandwidth; continuing improvements in energy capacity, alternative energy sources and energy harvesting. The need for CPS technologies is also being *pulled* by cyber-physical system vendors in sectors like aerospace, building and environmental control, critical infrastructure, process control, factory automation, and healthcare, who are increasingly finding that the technology base to build large-scale safety-critical CPS correctly, affordably, flexibly and on schedule is seriously lacking.

The great opportunity opened by CPS for industry, business and economy in general, cannot be missed. As stated in [28], "The cultural change must take place mainly on the provider side. The willingness to make radical changes in business processes or even new business models can only come from here. If the market forces them, then it is usually too late." Inter- and trans-disciplinarily, value creation and innovation in corporate networks and in business ecosystems are required that face the above challenges and take a leading role.

---

[8] Prosumer = **pro**ducer and con**sumer**

3 Homogeneous Description of Heterogeneous Systems

CPS cover various sectors and application domains. Examples of CPS exist in medical devices and systems, aerospace systems, power grids, transportation vehicles and intelligent highways, defence systems, robotic systems, process control, factory automation, building and environmental control or smart spaces. [47]

3.1 Context of CPS

Each CPS depends on specific contexts, as exemplified by Figure 1. Two types of context exist a *direct context* which influences the implemented system in its runtime behaviour through an interface and an *indirect context*, which influences e.g. the design, implementation, requirements, architecture or development process of the CPS, but has no direct interface to the system. The set of direct contexts of a system are a subset of the indirect contexts.

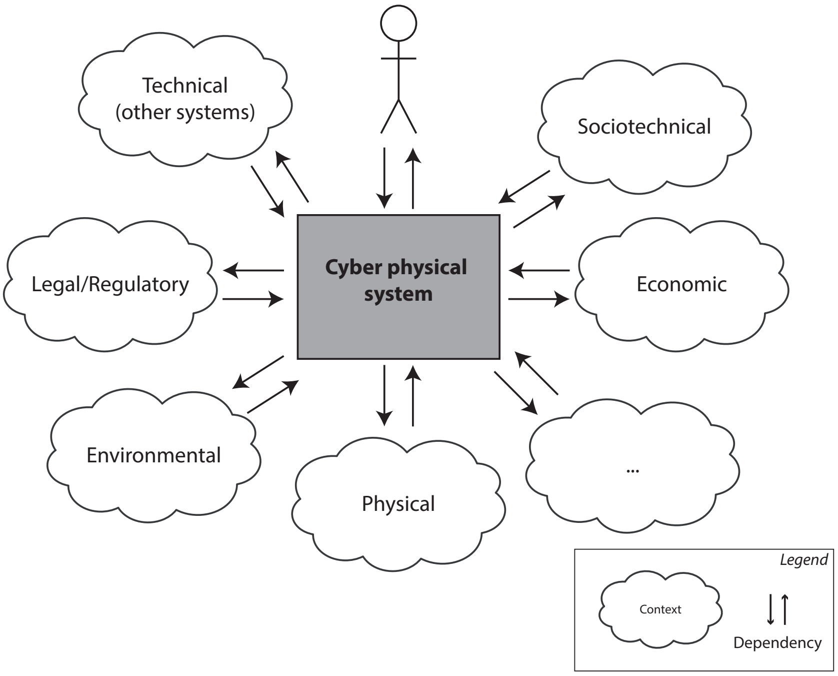


Fig. 1. Cyber physical systems development with respect to different contexts

An example for a direct context would be the physical context. CPS perceive this context through sensor measurements and interact with it through actuators. A car for example has sensors to measure its velocity, the intensity of the ambient light outside the car or the distance to a car driving ahead. Systems inside the car decide on the basis of the sensor data to accelerate or decelerate through a change in gear and engine speed, to adjust the lighting through an adjustment in the intensity of the headlights or to send a braking signal. In order to choose the right decisions at any time the system has to consider the physical context of the car. This physical context needs to be explored and understood to implement a system which shows the intended behaviour. Direct contexts are often mentioned in the system's documentation, as they are important to understand the runtime behaviour of implemented CPS.

Indirect contexts are for the development of most CPS as important as direct contexts. Indirect contexts can have an influence on the system's rationales, design decisions, requirements, architecture, coding, implementation, deployment, testing, specification, development process, but do not interact with the system's interface. For example, the legal context (e.g. legislation) is able to influence a CPS through laws, which could enforce certain requirements, for example on privacy, security, integrity, real-time properties or the design of the graphical user interface. On the other hand, the design of a system has a direct influence on the legislative. Laws have often to refer to e.g. the typical design of systems in a domain (e.g. the electrical power system), their domain-specific terms and implementation standards. As laws do not directly interact with the system's interface, they are clearly an indirect context of the system.

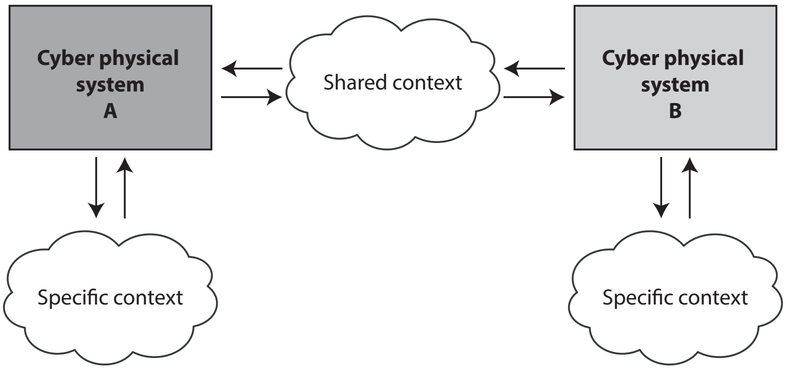


Fig. 2. Cyber physical systems may share the same context

With the on-going dissemination of CPS, many systems that were originally developed in separate application domains overlap in their respective contexts (Figure 2). This share of context can lead to an indirect interaction of the systems through their respective context. For many systems a deeper understanding of this shared context is not relevant or of minor importance.

With the interconnection of systems to so called systems of systems, the analysis of the common context allows to ensure interoperability, to specify safety properties or to create new systems that employ the shared context of certain CPS do derive new functionality.

Nevertheless it is not obvious if different systems overlap in their system context. The information on the context of a system may be hidden in the system specification, documentation or implementation. The usage of different terms, notions or description mechanisms complicate the problem to retrieve the information on systems context.

Since systems can act as the technical context of other systems (Figure 3), it is favourable to describe the architecture of different CPS to relate different parts of these systems to each other, to reuse solutions between different application areas or to show the interfaces that connect CPS to each other.

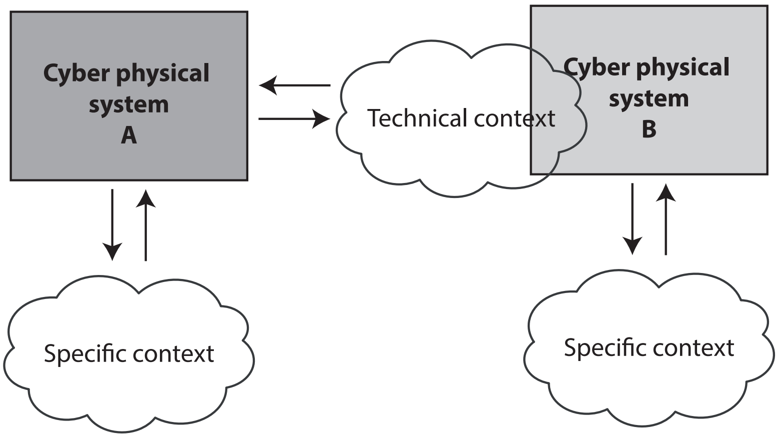


Fig. 3. One CPS may act as the technical context of another CPS.

A reference framework for cyber physical engineering defining a common context for different CPS may be the solution for the described problem. It should include a homogeneous description mechanism for both - classes of systems and their context.

To understand the contextual domain elements a reference framework has to cover, we want to illustrate contextual elements which were analysed by the study agendaCPS [23].

The study highlights the open and pervasive nature of CPS. Using a schematic representation (Figure 4) of two application domains, namely mobility and health, it characterizes the participants in a domain: Systems, users and stakeholders, as well as their interaction relations with respect to controllability, precision and predictability of their behaviour.

The study separates three so called action fields:

- **Controlled area (1)**: This area includes traditional closed-end embedded systems of a field of application (for example, the heating of a building), which are characterized by controlled communication and interaction with the environment. The reliability and predictability of the system behaviour are guaranteed while correct operation.

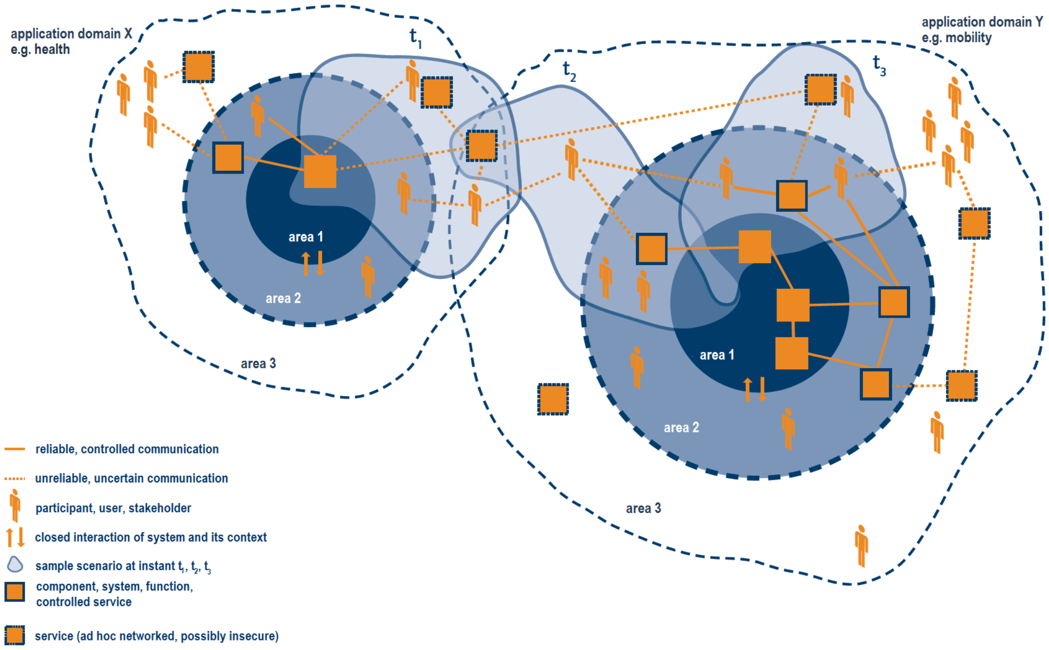


Fig. 4. Overview of the domain structure on the level of predictability of the behaviour of the involved systems and people (Source: Acatech)

- **Defined area (2)**. Systems and components of the application area cooperate within this area. They behave in the specified manner for predetermined and delimited usage situations. Their behaviour is uncertain in unpredictable, non-compliant situations. The communication within the ranges (1) and (2) is controlled and targeted.
- **Demarcated area (3)**: This open environment contains users, actors, group of actors, systems, services and data, which are related to one or several application areas. From the perspective of the inner regions, they are characterized by reduced reliability in terms of their source and their communication. Also, their contexts, goals and behaviours cannot be clearly classified and interpreted. It is decisive that users and components from the environment or services from the Internet interact with each other in an ad-hoc manner. The components of the open, connected world use components of the inner area as their context. In this way, new usage potential as well as new risks can emerge.

In the perspective of the study there are several classes of interaction between different contexts which have to be represented by a reference framework. It can be concluded, that typical elements of a reference framework include actors, roles, group of actors, systems, services, data and communication interfaces (not limited to systems).

These elements, which can be found in nearly any system of any domain, cover different application contexts. Depending on the type of an actor, a role or a group of actors it can exist for example in the user context of a system, in a legal context or in an economic context.

3.2 CPS Elements for a Reference Framework

For each CPS it is of paramount importance to define its system border, but also its context border. Considering the definition of a system border, it is necessary to border it from the system and the irrelevant part of reality. Through system and context borders, the system context is defined. It covers all aspects which are relevant for the requirements and cannot be designed during the development of the system. Therefore, two processes are defined:

(i) **Defining the system border**: This process determines which aspects will be covered by a CPS (the system's scope) and which aspects are part of the environment. All elements within the scope can be influenced and designed during development of a system. Within the border, there might be a hardware and software based entity. Interfaces to external systems form the border and offer points for interaction. System borders are normally described by views that model the systems internal structures, data elements, interactions, and operation.

(ii) **Defining the context border**: This process determines the separation between the context of the system and the irrelevant reality which has no influence on the CPS or its development. Entities within the context are typical sources for requirements. This might be stakeholders, external systems, existing processes, events and actions or legal definitions.

The description of CPS domains such as electrical power systems, transportation systems, healthcare systems or automotive systems includes the same processes. Nevertheless the two processes have to be amplified to classes of systems within the CPS domain, to derive a description of CPS domains and their context. This description can be included into the reference framework as a common description for the systems within domains and their respective context.

Defining the CPS domain border

The CPS domain border defines the common border of systems that belong to the CPS domain. The definition of the domain border is based on the empirical analysis of a set of proven domain-specific systems and their architectural properties. We suggest pattern identification, abstraction and generalization of individual architectures to define the common CPS border. Separation is one of the key concepts for any successful pattern. This concept allows the pattern documenter to separate the idea from the reality, capturing more generalized concepts when documenting the pattern. Another valuable pattern concept within the software community is the rule of three, first identified by Ralph Johnson [52], which states that a pattern is not a pattern unless there are at least three independent, observable applications utilizing the proposed pattern as part of the solution. Another key concept is abstraction the notion of removing detail from something complex to make it simpler to understand. Said differently, abstraction is the extraction of the essence to make something providing an architecture baseline and an architecture blueprint. These concepts can be applied to define the CPS domain border.

Defining the CPS domain context border

Typically, the system context is documented by using use-case, data flow or class models. All three illustrate the different actors in the environment and their interaction with the system. For a complete documentation, different forms are used, highlighting changing perspectives on the system and its environment. Within a CPS domain these use-case, data flows or class models have to be amplified to a domain-wide reference context. Therefore we again suggest to empirically rely on the knowledge gained in the development of proven systems within the domain.

We assume that the analysis of the context and system borders of several CPS domains will include common elements that can be used for more than one domain. Both, the border of the CPS domain and the domain context border have to be described by a homogeneous description mechanism to ensure an efficient collaboration between separate CPS domains. The identified elements can be used within the reference framework to model cross-domain dependencies in terms of contexts of CPS.

We use our experience in two cyber physical domains, namely *Smart Grid* and *Automotive* to show the main elements of the domains and their context.

4 Challenges

Associated with the materialization of the vision CPS, there is a series of challenges of different nature. We present a number of them, grouped in the categories technical, organizational, and social. Spanning over these all, there is the matter of how to support the diverse interest groups, which is briefly discussed at the end of the present section.

4.1 Technical Challenges

There is extensive literature dealing with the challenges associated with the (a) modelling, development, realization, (b) validation, verification and test, as well as (c) maintenance and evolution of CPS. At least two different kind of technical challenges can be identified. On the one hand, precisely because of the evolution that lead to the definition of these novel systems, CPS are not conceived and implemented "from scratch": while embedded systems combine sensing and actuating mechanisms with (simple) computational power, CPS go a step further and network embedded systems. The resulting systems open a wide range of possibilities, as we are witnessing, e.g., in the realm of telecommunications. At the same time, this means that there is an enormous amount of legacy systems, that should be made "CPS ready" in order to be meaningfully interconnected with other systems, i.e., to take advantage of and offer services.

On the other hand, the above mentioned networking can only take place if there is a communication protocol that ensures, among others, that the information conveyed is correctly interpreted, that its integrity is preserved, and depending on the circumstances that it is only accessed by its addressee. This communication must likely be immersed in a complex platform, that provides much more than just a vehicle of information. It must manage a huge information volume, while taking care of its integrity and confidentiality. It has to mediate between systems of inherently diverse nature. A proposal for the realization of these services is proposed by the middleware Chromosome (see [13,29]), that 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.

4.2 Organizational Challenges

By organizational challenges we understand the matter of standardization, a sine qua non for the infrastructure of the above section to be widely useful, as well as the issues associated with regulations and legislation.

Regarding liability, for instance, a debate is imperative that clarifies governance as well as normative and regulative aspects, and determines the authority of humans on CPS and vice versa; cf. [37, Sect. 3.2]. Many constellations, settings and risks can be foreseen and anticipatory resolved. Some others, as the evolution of technology teaches us, will have to be settled on the fly, or in some cases even retrospectively.

With respect to standards, there is a tension between competition and collaboration. For instance in the mobile telecommunication realm, the market has witnessed an invasion of so-called apps, that are independently developed and can nevertheless cooperate with each other. They are, however, incompatible if downloaded from the wrong provider (which puts apps at disposal for another platform, different from the one at hand). This is a similar evolution underwent by PCs, operating systems and software, be this purchasable, public domain or free: standards and monopolies are not to be treated as one and the same concept.

Hence, some market regulation mechanisms may be necessary. Furthermore, also a clear policy may prove promotive, not only in order to avoid unfair competition, but also to enable progress as, e.g., it is often done towards the switch to renewable energy generation.

Related to the matter of standards, there is the need of a common understanding among experts from different realms, for their systems and solution to be successfully combined. This must be the case even if the systems that are supposed to cooperate have been designed and developed following totally different approaches (at it might be with solutions in the domains of, e.g., health and of transportation: as soon as one thinks of autonomous cars and ambulances, there must be a coordination taking place). A single reference (meta-)architecture for heterogeneous domains might be a good idea; but it also can be deemed too restrictive, and in the best case can norm future developments, not past ones.

4.3 Social Challenges

The social challenges we address here can also be termed human factors.

The internet transformed how and where information is stored and accessed, the way people interact and communicate with one another, including, e.g., how products are bought and sold, services provided, etc. In a similar fashion, CPS transform how users interact with and control the surrounding physical world. Those systems are expected to operate dependably, safely, securely, efficiently and in real-time (the list is not exhaustive). A scientific and engineering CPS discipline should advance the conceptualization and realization of future societal-scale systems, supported by an analysis of the interactions between engineered structures, information processing, humans and the physical world. In particular, the engineering of these novel systems must take into account, among others, the availability and the constraints associated not only with their cyber and/or physical components, but also with the human operators; see [47].

Note that the operator not necessarily is a highly trained engineer. The CPS vision encompasses increased support and care of the aging population. The infrastructure put at disposal of patients with age-related, chronic diseases can be substantially improved by means of CPS, so that the elderly can comfortably stay at home longer.

In the automotive domain, and due to the complexity of the environment and to detection accuracy as well as to legal constraints, completely autonomous, self-driving cars will not conceivably roll in the next future on public streets. The benefits, therefore, of CPS in cars will by and large depend on how human drivers interact with them

The same can be asserted of, e.g., the interaction of physicians with their healthcare infrastructure. Moreover, there are not only the different levels of education and/or training, there are also the generational leaps, cultural backgrounds, social and wealth discrepancies, and also the so-called dropouts that cannot afford or do not desire any contact with the new technologies. In general, thus, usability of CPS poses a variety of challenges involving computer-human interaction and interface design.

Most of the existing research on human factors has only studied how external factors, such as road signs and warnings from driving assistance systems, affect drivers and traffic. There is, therefore, the need for research of those issues *a priori*, i.e., how to consider human factors at the design stage of systems; see [38].

4.4 Education and Training Support

Spanning the above perspectives, an additional challenge is posed by the training and education of present and future generations. [9] This applies not only to the operation of the systems, but also to their development. When speaking of CPS operation, meant are both the operators of more or less sophisticated plants (like, e.g., power plants) as well as average consumers (as, e.g., elderly patients who represent a non-insignificant proportion of the intended beneficiaries of the smart health vision). This goal can achieved by means of

---

[9] Sometimes progress in general, and technological progress in particular, trigger emotional debates that are more feelings (of, e.g., nostalgic nature) based on domestic observations than hard facts. This still happens with respect to television and its supposed harmfulness, to “shoot ’em up” computer games and their supposed link to running amok, and to online gaming populations and the loss of “offline” sociability. What is more, although some studies have claimed to show a link between video games and violent or aggressive behavior, most research in this area has been flawed; see [46] and also [36,35]. The technology, as such, cannot be perceived as injurious (unless explicitly designed to that end); its use can, though.

workshops, seminars and individual coaching. Be it for users, operators, designers or developers, a common (i.e., homogeneous) terminology is of utmost importance.

Parallel to the evolution in the realization of CPS, the institutions of higher education should update their syllabus. That is, the insights gained in the industrial realm need to find their way back to the university, where their fundamental challenges can be tackled in such a way that a sound theory of CPS development can come about.

5 Experiences from the Smart Grid Domain

Smart Grid is a CPS par excellence. It combines the physical power system with information and communication technology creating a large, distributed system of systems, which comprises all expected characteristics from the definition in [23].

5.1 Domain Overview

The term *smart grid* was originally introduced to highlight the development of the electrical system enhancing the capabilities of the 20th century power grid. Traditional power grids were designed to transport electricity from a few central generators to a large number of customers, as shown in Figure 5. The operation of the system was organized by few companies having a natural monopoly in their area. This is a clear separation of responsibilities and provides a convenient engineering environment of the system.

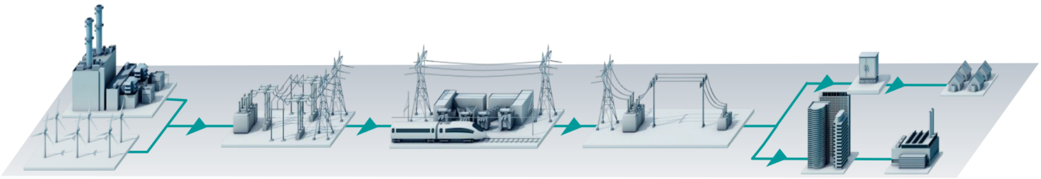


Fig. 5. Traditional power system design: centralized and unidirectional (Source: Siemens)

These days, political goals, technological progress and economic pressure require the modernization towards a more advanced power system. The desire to generate power from renewable and distributed energy resources (DER) challenges current power system design and changes the paradigm of system operation. As renewable sources depend strongly on volatile factors (e.g. weather), production cannot always be adjusted to the consumption. To operate the system in a safe and reliable way, consumption needs to become more adaptive, storage capacities need to be exploited and advanced energy management systems need to be developed to integrate all new system capabilities. Smart grid systems promise to handle these challenges with the use of information and communication technologies (ICT) enabling bidirectional information and power flows within the system. The ultimate smart grid is a vision. It is a highly adaptive and automated system consisting of interconnected complemented components, subsystems, functions and services for power generation, transmission, distribution, consumption, monitoring and control [8,24]. The goal is to achieve a system, which is efficient, clean, reliable, safe, secure, sustainable, able to react (proactively) to wide ranging conditions and events, and adopt the corresponding strategies. The smart grid system design is shown in Figure 6.

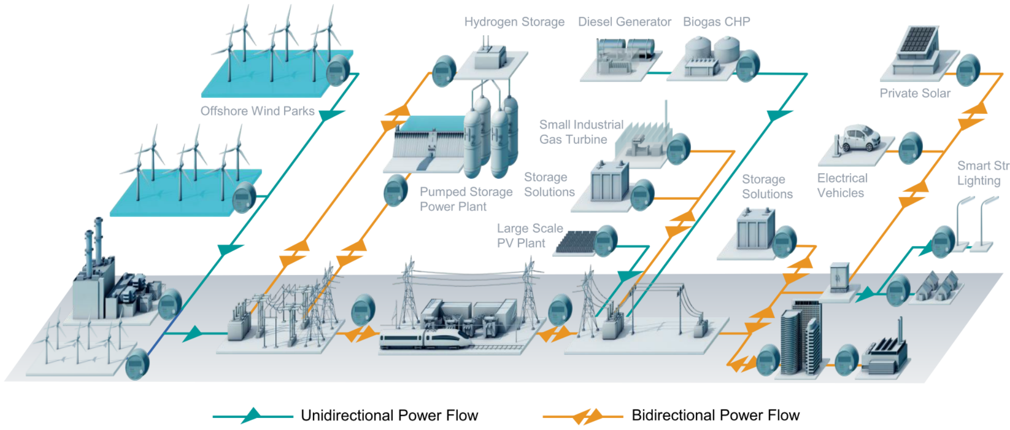


Fig. 6. Smart grid system design: distributed and bidirectional energy balancing (Source: Siemens)

5.2 Research Activities

The effort for smart grid development projects was tremendous in the recent years. Pilot projects, research programs, field test, standardization and integration activities were supported by governments and carried out by academia, industry and research organization. Within the European Union a JCR Study [25] has identified 281 projects up to 2012 accounting for a total investment of 1.8 billion EUR. The Smart Grid Information Clearinghouse [10] observed similar investment numbers for smart grid efforts all over the world. This large amount of projects provides an excellent example to study smart grid engineering challenges and identify particulars and commonalities for a CPS Engineering Framework.

Beside the large number of smart grid projects we can observe many system integration efforts and standardization activities all over the world. The major international, national and associations activities are:

(i) IEEE P2030 Smart Grid Interoperability Framework [30]
(ii) EU: Mandate M490 Smart Grid Reference Architecture (SGAM) [18]
(iii) US: NIST Framework and Roadmap for SG Interoperability [45]
(iv) Germany: BMWi E-Energy Program [11], VDE/DKE Roadmap [64]
(v) China: SGCC Framework and Roadmap [60]
(vi) IEC Smart Grid Standardization Roadmap [58]

Further countries, such as Japan's Smart Community Alliance Activities [44], Korea's Agency for Technology and Standards (KATS), India and Brazil have started their own standardization activities [64] and more regional roadmaps exists (e.g. Austria [4], Denmark[3], Spain[20] and UK [16,17]).

---

[10] http://www.sgiclearinghouse.org

A comparison of the approaches and standardization activities can be found in [53,63]. Those studies identify the most accepted standards and indicate open topics, which are insufficiently addressed by the community.

Technical research on smart grids can be classified to three major categories [19]: the **smart infrastructure system**, the **smart management system** and the **smart protection system**. Each of these research areas covers different activities and disciplines.

- The smart infrastructure system addresses technical components and their integration to the infrastructure. It corresponds to the IEEE P2030 reference architecture distinguishing three subdomains: a *smart energy subsystem*, which is responsible for effective energy generation, transport, distribution, and consumption; a *smart information subsystem* for data metering, monitoring and management; and a *smart communication system* responsible for the communication among all the systems within a smart grid.
- The smart management system addresses research activates towards more advanced services and functionalities. Those services and functionalities leverage the capabilities of the smart infrastructure and enable new solutions for grid operation. The functions follow different management objectives, e.g. demand response, reduction of emissions, data aggregation, and address smart grid related management methods and tools, e.g. optimization algorithms.
- The smart protection system covers all research activities towards safety, reliability, failure identification and localization, self-healing, resilience, and also security and privacy.

Beside technical oriented research many interdisciplinary challenges exist to reach a fully deployed smart grid system. Major challenges include political, legal, economical, socio-cultural, and human-oriented aspects. Political and legal factors are basically to what degree the government intervenes into the system. They include acts of parliament and associated regulations, international and national standards, local government by-laws, and mechanisms to monitor and ensure compliance with these. These factors affect the requirements for products and services offered by companies, market environment and facilitation of technological progress. Economic factors provide an environment for obstacles, opportunities and innovation based on economic growth, interest rates, capital investments and determe how business operates. The acceptance and impact of technology depends on socio-cultural factors. In particular the usage of new technology and services requires people to accustom themself to new technology and processes, especially when it comes to disruptive change of the habitus. A CPS reference framework needs to capture those challenges as early as possible, and integrate them in the development process.

5.3 Smart Grid Observations

The large number of smart grid projects developed a plethora of systems including pilot projects, demonstrators, field tests, approaches for optimization, modelling, verification, simulation as well as supporting guidelines and frameworks for the engineering of smart grid technology. The results include many different aspects of the energy system, involve different stakeholders, include new components, functions and data structures and use different technologies, concepts, terminology and infrastructure.

The sheer amount of existing systems, their different architectures, different level of abstraction and a varying use of terms and language complicate the comprehension and comparison of different solutions. However, for many tasks that need an abstract system comprehension such as the design of a reference system architecture, regulatory frameworks, the development of standards, legislation, discussions on a national or international level, market design or the collaboration between different disciplines, a common domain understand and language is of great benefit. Therefore, more effort should be invested to define a comprehensible abstract view of the system including its context and boundaries, a common language for collaborative interaction between the disciplines, standard activities beyond national boundaries, and opportunities for dynamic and spontaneous collaboration.

Smart Grid Reference Architectures

A good example pointing out important requirements for the CPS reference framework is the development of the smart grid reference architecture. In our point of view, a *reference architecture* is a reference model that captures architectural knowledge about a class of systems. Based on individual system architectures, it encompasses central architectural concepts of the class and their interrelation. It must be supported by a unified, explicit, unambiguous, and widely understood domain terminology. The comparison to other definitions of the term reference architecture is presented in [32].

A popular smart grid reference architecture is the **IEEE P2030** standard. It was developed to provide a common understanding, terminology, definitions and guidance for design and implementation of Smart Grid components and end-use applications for both legacy and future infrastructures [30]. IEEE P2030 offers three major perspectives, the power systems perspective, the communication technology perspective and the information technology perspective. Furthermore, each perspective is comprised of seven domains: bulk generation, transmission, distribution, service provides, markets, control, operations and customers. Every domain consists of a number of entities, which are logically connected with interfaces.

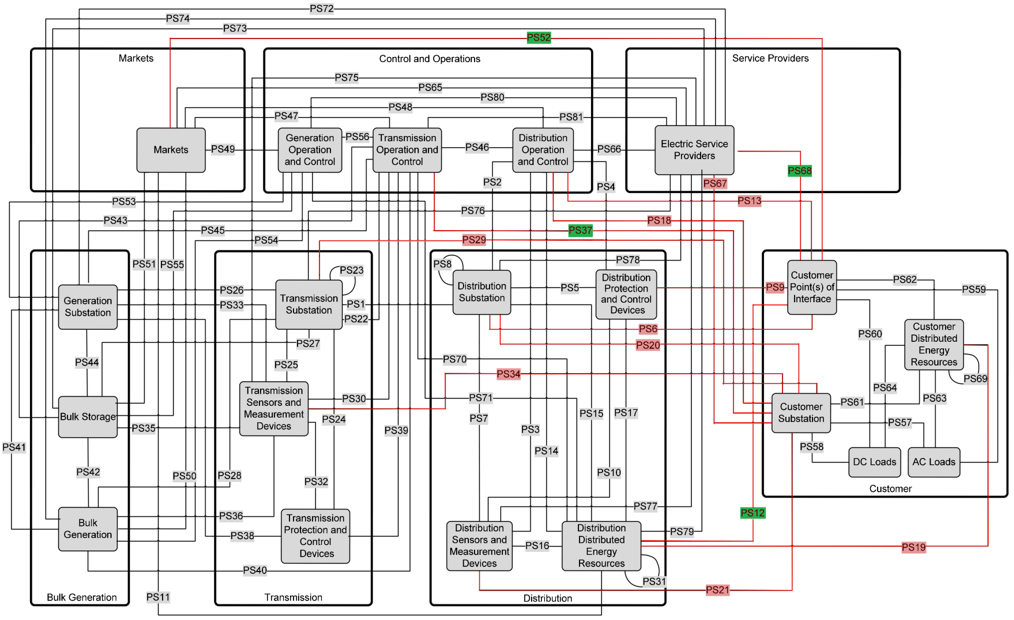


Fig. 7. IEEE P2030 PS-IAP with highlighted connections to the customer domain

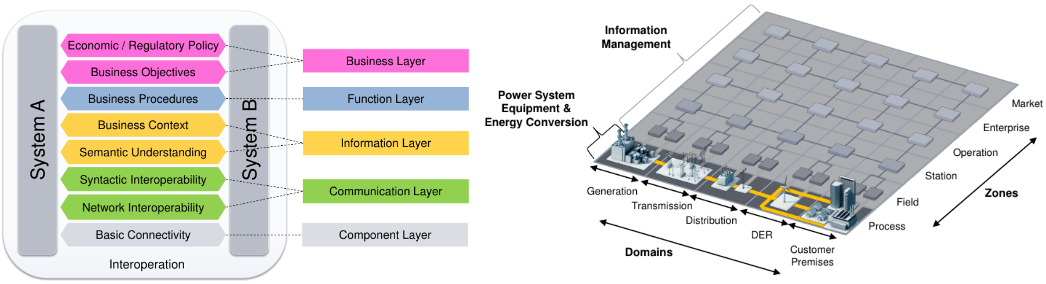
The IEEE P2030 power system perspective is shown in Figure 7. The interfaces to the customer domain are highlighted. The connections are specified by the standard as exemplarily shown in Figure 8 for the green highlighted connections. The communication and information technology perspective follow the same structure.

Interface	**Entity 1**	**Entity 2**	**Comments**
PS12	Customer Point(s) of Interface	Distribution Distributed Energy Resources	Provides for aggregated customer information and distribution system DER control directly to customer. Provides a means to locally balance generation and loads. Interfaces include those for control and monitoring.
PS37	Customer Substation	Transmission Operation and Control	Provides for transmission operators to monitor and control customer substations. Interfaces include those for control, monitoring, SCADA, reporting, and telephony.
PS52	Customer Point(s) of Interface	Markets	Provides for optimization of distributed generation, storage, and load control (i.e., demand response) on the customer domain. Interfaces include those for control, monitoring, and reporting.
PS68	Customer Point(s) of Interface	Electric Service Providers	Provides for monitoring information and control of customer generation, storage, and loads. Interfaces include those for monitoring and control.

Fig. 8. Exemplary PS-IAP interfaces (Source: [30])

IEEE P2030 seems to be a promising reference architecture for standardization of interfaces, however, there is no evidence how well existing smart grid concepts can be represented in this approach, since links between different perspectives and detailed specifications are not presented. Therefore it is unclear if this reference architecture is feasible for analysis with respect to functionalities, envisioned for the smart grid and their manifestation within the system.

European Smart Grid activities are working on other reference architectures. The most promising candidate seems to be the Mandate M/490, where the CEN-CENELEC-ETSI Smart Grid Coordination Group is requested to provide a reference **Smart Grid Architecture Model (SGAM)** framework [18]. Instead of perspectives and domains, SGAM uses a three dimensional structure with: five interoperability layers representing different abstraction views - business, function, information, communication and component; five domains representing the physical energy value chain levels - generation, transmission, distribution, distributed energy resources (DER) and customer premises; five zones representing the power management levels - process, field, station, enterprise, market. Furthermore, SGAM mentions cross cutting issues, such as security. The concept's elements are shown in Figure 9 and the complete SGAM Framework is shown in Figure 10.



(a) System Interoperability Layers (b) Domains and Management Levels

Fig. 9. Elements of the SGAM Framework

The reference architecture has three major objectives: to represent the main smart grid elements with all major stakeholders, to support a variety of different approaches and to provide the basis to search for required standards to build an interoperable system. Envisioned use cases are employed for the identification process of relevant standards. The SGAM method starts on the bottom layer for components and continues on the business layer to include organizations and stakeholders. Afterwards the identification is carried out from the business layer down to the communication layer.

Both reference architectures, IEEE P2030 and SGAM, are designed to facilitate interoperability by the definition of standards for interfaces. They represent in their view an abstraction of best practice, condensed from numerous case studies over an extended period of time, followed by more case studies to refine and evaluate the proposed reference model. There is noth-

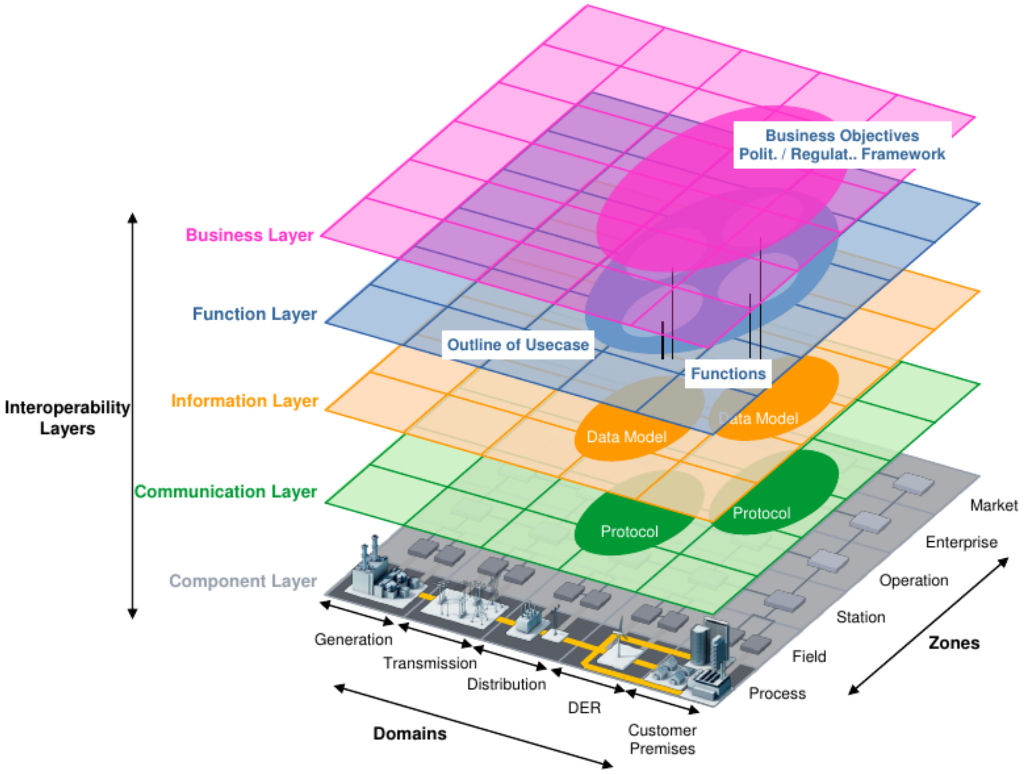


Fig. 10. Complete SGAM framework and correspondent objectives

ing provably correct about reference architectures; they derive their relevance from the slice of practice they cover. Consequently, the validity and feasibility of a reference architecture has to be shown for the intended class of systems, otherwise its chance of acceptance is low.

The effort on reference architectures is an accumulation of knowledge about a certain domain. The standards derived from the reference architecture contain often data models, communication protocols and cover even dynamic interconnection of system components. However, they are still weak in terms of interconnectivity between different levels of abstraction and the representation of components and functions into the system. The CPS reference framework needs to fill this gap and provide the capability to classify and interconnect components over different domains.

Legacy Systems

Transferring project results such as reference architectures, modelling concepts and research prototypes into well-established industrial context is not easy, especially if the life cycle of system components is very long and reliability is crucial. Developers of physical power system components consider life cycles with more than 20 years with only little manual interaction with system operators and maintenance engineers. Confidence that such components are reliable and safe is therefore of paramount importance. A CPS reference framework needs a proper integration of legacy systems, which will be present for decades in smart grids.

Over the years smart grid technology will evolve and provide new opportunities and services as an infrastructure. Components will be required to be easily integrated into the new environment with the capability for on-going maintenance. A reference framework should therefore support the development of such components and provide the capability for their integration into a real CPS. Therefore a set of methods and artefacts is required for each of the CPS framework elements.

6 Experiences from the Automotive Domain

A typical metaphor for the variety of cyber physical systems that will emerge is the *smart car* driving through and interacting with a *smart city* that is powered by a *smart grid*. In traditional automotive systems engineering, the vehicle is seen mainly from a technical, component oriented perspective. Today, with a development towards piloted and, in the near future, highly autonomous driving capabilities, the car becomes an intelligent system that is tightly embedded into its physical environment ant the supporting infrastructure. So, to enable such an ubiquitous scenario, the car itself has to become a *CPS* in conjunction with the surrounding CPSs.

6.1 Traditional Automotive Architectures

Figure 11 displays an example topology of the ECU network of a current upper class vehicle. In this topology, the logical architecture of components that define the behaviour of the system resembles the technical architecture - the network of electronic control units (ECUs).

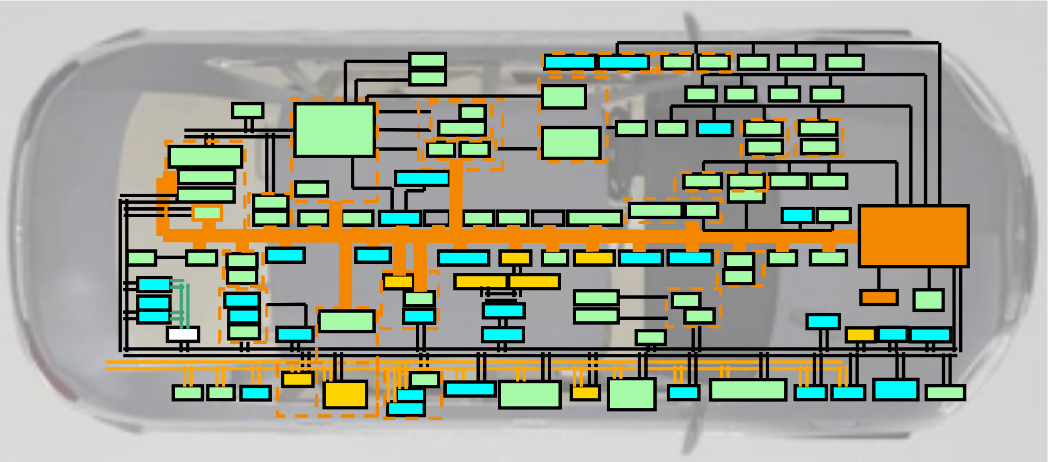


Fig. 11. Traditional technical automotive architecture

Although the presented architecture might very well provide the customer with advanced functionality such as adaptive cruise control (ACC), automated parking or predictive braking, there is limited potential for further enhancements. The electric/electronic (E/E) architecture of a vehicle constructed according to the AUTOSAR standard [5] is the result of a structure that is historically grown and has been developed further over the last decade. Because of this history, there are five fundamental functional domains:

(i) **Chassis:** Functions and components concerning the base structure of the vehicle, such as brakes and suspensions. This emerged from the time where the chassis of a vehicle was the supporting structure, often made from a metal pipe frame. In modern vehicles, the physical properties of body and chassis converge as the vehicle usually contains a self-supporting body. The chassis domain remained as a container for vehicle features formerly associated with the chassis.

(ii) **Body:** Lights, windows and comfort related functionality. As stated before, from a physical perspective, it is not obvious to distinguish between body and chassis any more. Although, the domains divide vehicle features in more driving-related features (suspension, brakes, steering) allocated to the chassis domain and secondary functionality (light, comfort) associated to the vehicle body

(iii) **Powertrain:** Engine control components. As the main component of a vehicle is the engine, a dedicated domain for engine control and its supporting components was established. With the introduction of alternative propulsion systems, such as hybrid, hydrogen or solely electric engines or combinations, increasing complexity is expected in this domain. Also, the strict separation of braking, steering and propulsion control inhibits the development of more sophisticated stability control systems that are able to cover positive as well as negative acceleration forces.

(iv) **Passenger and Pedestrian Safety:** Active and passive safety features such as airbags. With the emergence of automated or autonomous driving features, the aspect of providing active safety with respect to the passenger as well as the surrounding environment gains a much higher impact on the overall behaviour of the vehicle.

(v) **Infotainment and Telematics:** Non safety related functionality, mainly the entertainment system. While today's telematics solutions allow for dynamic navigation routing via traffic information, future vehicle will come with a much higher level of interconnection between the infrastructure and other vehicles participating in traffic.

There are several drivers that lead to the establishment of such a domain structure. Mainly, the domain structure is an organizational entity that allows for grouping and encapsulation of vehicle features according to the physical impact on the vehicle. This simplifies the development process as only a subset of the overall vehicle functionality has to be taken into account when developing or modifying a specific feature. Each domain has dedicated communication buses that connect the application specific ECUs. This also helps to prevent propagation of errors, supports the reduction of bus traffic due to domain-dedicated communication infrastructure and enables separation.

6.2 Evolution of Automotive E/E Architectures

Unfortunately, with the introduction of more sophisticated functionality such as advanced driver assistance systems (ADAS), it becomes more complex to maintain such a domain structure. Those functions have to be realized across multiple domains and therefore undermine the domain separation. There is already an observable degradation in architecture quality as inter domain traffic increases. This leads to very complex architectures as suddenly all domains have to be taken into account when developing a new feature as inter domain traffic will play a huge role in most cases.

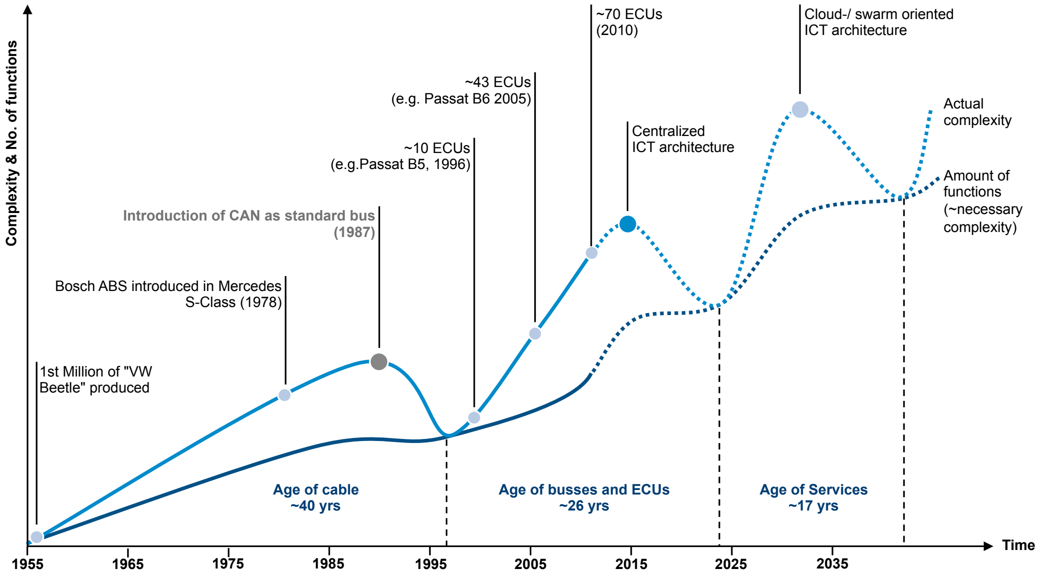


Fig. 12. Available amount of functionality vs. architecture complexity [7]

A recently conducted study [7] investigated the efforts to be taken to develop and maintain such an architecture with respect to the amount of functionality realized. The results are shown in Figure 12. The current development leads to a point where it is not feasible, neither economically nor technologically, to integrate further functionality into the system. At that point, a revolutionary step away from the traditional approach has to be taken. The study proposes the introduction of a centralized automotive ICT architecture to overcome the limitations and to enable sophisticated new functionality.

The future scenarios see the isolated intelligent vehicle with a centralized ICT architecture only as a preliminary step towards a completely networked system of vehicle CPSs where all vehicles are connected and, together with infrastructure systems, form an ubiquitous cloud architecture where the allocation of functionality or actual control over the behaviour of a single node in the system is transparently distributed. Section 6.3 introduces the challenges of a future automotive architecture.

6.3 Challenges of future Automotive Architectures

Aside from the properties of future automotive architectures, several questions with respect to the specification, integration, validation/verification and maintenance of such systems are remaining open questions:

- How can we model the application context/domain?
- How do we know which features interact?
- How can we assure functional integrity in a Plug and Play scenario?
- How can we assure functionality- and quality-related properties?

In the next section, we distil an overview on domain-specific and common properties of CPS.

7 Particulars and Commonalities

We analysed two CPS application domains in the Sections 5 and 6, that are respectively energy smart grid and automotive. We report in this section particulars and commonalities of CPS in these domains respect to other domains.

7.1 Smart Grid

The smart grid vision, as described above, is a highly adaptive and automated system consisting of interconnected complemented components, subsystems, functions and services for power generation, transmission, distribution, consumption, monitoring and control. The specific domain of power systems constrains the functionality to power equipment and different services to maintain and manage the equipment. But at the same time smart grid provides an ideal infrastructure with many extension points for all kinds of CPS systems. For example a smart grid connected house includes a variety of components, such as coffee machines, water pumps, heating and cooling equipment, which are connected to smart energy management systems. These systems gather a lot of data through sensors and actuators, which can be provided to external systems. A smart meter can measure the consumption of a coffee machine, actuators can switch on/off lights and power sockets, batteries can save energy for emergencies. This information can be made accessible to other systems.

7.1.1 Particulars

The main objective of a smart grid is the modernization of the traditional power system to integrate a large number of decentralized renewable energy sources. A lot of smart grid research is therefore focused particular problems such as the physical *integration of volatile, decentralized power injections* into the grid and related *control strategies*. A fundamental physical constraint of power systems is that the power must be balanced at all time. The increasing *inhomogeneous energy generation* and *stochastic energy consumption* of consumers needs to be matched in safe and reliable way. Important tasks are therefore to investigate, *how to coordinate many individual energy management systems in a proper way*, *what data is needed*, *how is date communicated*, *which ICT architectures* and *control strategies* ensure a stable network operation. Further technical particulars of smart grids include: *a cost effective extension of the sensor network of the power distribution system*, *integration of cost effective storage capabilities*, *integration of electrical vehicles to the power grid* and *a high level of automatizing of a large number of components.*

Beside technical particulars, political regulations such as the *nuclear phase-out* and *economic requirements* are important factors of smart grids. A good example is the *market design*, which represents the regulation framework for trading energy and related services. The current market design prefers large and reliable power generation and consumption. Smaller stakeholders cannot participate. This requires an adapted market design in future, or an aggregation capability, as the majority of smart grid components will be smaller in size with a reduced reliability.

7.1.2 Commonalities

The experiences from smart grid offer a large variety of topics, which can be transferred to other CPS systems. We show a list of important topics, which are subject in other domains as well.

- *Standardization:* Smart grid offers a set of standardization activities, condensed from numerous case studies over an extended period of time. The approach can be transferred to other domains to create reference models and standards for interoperability of different systems.
- *Modelling / Simulation / Verification:* The evaluation of smart grid system design is mainly based on large simulation and modelling efforts. The combination of physical and ICT modelling techniques can be used improve the methods for other CPS domains as well.
- *Reduction of complexity:* Services and functions can be aggregated in smart grids in a cell-like manner. This can be used to form a hierarchical structure and reduce the overall complexity. A cell consists of all required functions/ sensors/ actuators to form an adaptive (autonomic) system. Cells offer the capability for dynamic composition and decomposition.
- *Market design:* An appropriate market design can be used as an incentive for collaboration. In smart grids the market offers the possibility to trade certain type of services and consume or provide resources. The mechanism can be used to aggregate services or resources and create new products in a CPS system.
- *Regulation / Legislation:* Benefits of renewable energies drives the political attitude, which in turn supports the modernization of traditional power systems. The modernization is accompanied by a set of regulation and legislation activities, which impose a set of requirements and a variety of opportunities for the new infrastructure. The activities provide a set of best practices for future CPS system regulation.
- *Finance:* The modernization of the power systems is strongly subsidized. Future CPS are can benefit from the developed infrastructure (e.g. smart homes or building automation), or use the subsidiary schemes to finance new applications and interdisciplinary collaboration.
- *Sensors / Actuators:* Many new sensors and actuators will be placed in smart grid systems. The sensors and actuators will become more energy

efficient and smaller in size. They will collect a wide range of information, which will be available for other domains. In addition sensors and actuators will come with plug and play capabilities, creating a new class of smart devices.

- *Big Data:* The large amount of new sensors and actuators will produce a lot of data. This requires new data collection and processing systems. As not every data point is required all the time, new publish / subscribe methods will be developed.
- *Communication Infrastructure:* The new infrastructure will provide new communication capabilities. With more sensors and actuators, which might even move in space, the communication will move more towards wireless networks. In addition better batteries and energy harvesting methods will provide a distributed network of more communication devices.
- *Reliability:* The power system is essential for our society. New methods towards adaptive, autonomic and self-healing systems needs to be developed for physical and IT systems.
- *Maintenance:* Regular updates are common in today's software systems. However, physical and embedded components with life-cycles around 20 years and undesirable downtimes cannot be updated that easily. Therefore new update schemes needs to be developed, which are suited for long life cycles.
- *Smart Control and Adaptation:* The major objective of smart grids is a stable supply of energy. Most of renewable sources are strongly dependent on their environment and current weather conditions. The consumption of energy should therefore adapt with respect on the local condition and consume power whenever it is available. This requires a context sensitive and predictive control strategy for smart adaptation. The methods for context sensitive information and prediction based control are important in other CPS domains as well.

7.2 Automotive

A centralized architecture as an approach for future vehicles defines a disruptive development for the automotive industry. Therefore, all parts of the development process are affected. The main objectives when redefining the car as a "Smart Car" CPS concern the disruptive changes in the vehicles E/E architecture and how this will affect all stages of the development and integration process. This adapted process has also to take into account that the vehicle is not an isolated system anymore, but is embedded into and interacting with its environment.

Nowadays the rapid increase of the integration of software and electronic components in vehicles determines one of the most interesting application fields for CPS. As already mentioned the communication inside a car, but also with the environment represented by other cars or entities, leads to a complex and dynamic use scenario. In a smart car the ECUs architecture will grow constantly, in its number of elements and functionalities. Therefore, advanced techniques for the design of CPS are applied in the automotive domain, together with consistent investments due to the intrinsic importance of this sector. It follows that particulars and commonalities of the automotive domain help to conceive a wide application for the possible solutions and open problems.

7.2.1 Particulars

Improving the comfort, the safety, the influence on environment, the performance are some of the main goals of the conceived cars. Moreover, the increase of the smart features of the vehicles and their integration in a net of other smart entities is at a first glance a secondary bunch of objectives, but the particular of this domain is that they can then improve also the first group of goals. Another particular aspect to consider, is the suddenly transformation of the automotive from a domain based on a dominant discipline to a multi-disciplinary domain. In fact, in the last years the electronic and the computer science engineering increase their role alongside the mechanical engineering. It follows the needs for proved support tools, which guarantee also compatibility for legacy systems. The safety has a particular consideration in the design, due to the still relative high numbers of grave and mortal accidents.

Besides these technical aspects, also the economic has special characteristics. Traditionally when an economy is making is industrial sector mature, the automotive market is one of the first sector to expand, which then draws also other sectors to a stable development. Therefore, in some circumstances this field can attract more easily, instead of other fields, investments and governmental support. At the same time, the automotive market is assuming the aspects of a global market. In fact, the international competition and the economic advantages in the high number productions, obligate the companies to conceive products for the most number of market regions.

7.2.2 Commonalities

Most of the challenges associated with the automotive domain are common to other CPS applications. Frequent properties and objectives of automotive applications are:

- *Encapsulation of behaviour:* Different aspects of system behaviour can be provided separately. This encapsulation in a modular structure improves the quality of the system performance as it limits impact on the overall system health in case of an error. It furthermore supports the development process ("divide and conquer"): complex systems can be split into smaller modules for development and testing.

- *Reactive behaviour:* The main objective of the system is to react on environmental excitation with a physical reaction. This means that the system is able to capture information from the physical world, gain knowledge by interpreting the data, process the information and to react in a suitable manner if a reaction is indicated. The objective of any CPS is to react on physical excitation with a physical reaction which is defined as reactive behaviour. This is not necessarily the only objective, but the defining objective of the system.
- *Distributability:* As stated before, today's automotive architectures suffer from a strong resemblance of technical and logical architecture. For developing a consistent architecture, it is important to establish a set of views on the system that enable the developers to cover all relevant aspects as proposed with the SPES matrix introduced in section 8.2. In an isolated system, this enables the distribution of functionality over several architecture components, in a future systems of systems scenario, functionality could be distributed over the whole super-system.
- *Reliability:* A vehicle operates in situations that will, if not handled properly, can cause severe damage and will endanger health and even the life of passengers or non-involved pedestrians. There can be problems on several levels. A software feature, although specified and implemented correctly, could malfunction due to problems in a hardware component. As more functionality is transferred to and realized by software, quality assurance mechanisms as stated in the ISO 26262 standard [31] have to be applied. Additionally, architectural patterns should encapsulate functionality to limit the impact of a hazard to a part of the system. Most CPS fulfil critical tasks that could potentially cause damage or harm people in case of an accident. Therefore, integrity and quality assurance play an important role in all CPS.
- *Support for multi-functionality:* Next to the control of propulsion, steering, braking and stability, modern vehicles have a large variety of additional objectives to fulfil. Those objectives can also be quality-related, such as traveling from A to B while utilizing the least possible amount of fuel to protect the environment. Suitable mechanisms have to ensure that that all objectives can be fulfilled and desired and unwanted interaction between the features is discovered and can be dealt with.
- *Runtime adaptivity:* The evolution cycles in most technological areas become shorter and shorter. In contrary, the lifespan of the average vehicle increases due to improvements in build quality and materials and the in-

troduction of more intelligent control systems that protect the engine and associated aggregates. Future hybrid or electric vehicles might provide an even longer lifespan as an electrical engine is virtually wear-free. Therefore, the vehicle's functionality needs to be "updated" from time to time to keep up with new requirements, either legislative or customer driven. Such "Plug and Play" capability with respect to functionality and architecture components needs to ensure the preservation of behavioural integrity as the introduction of a new functionality is not allowed to undermine the reliability of the system. Other kinds of CPS are often applied in environments where they cannot just be switched off for maintenance but have to be serviced, configured and updated while in operation. This is especially relevant for CPS that are essential parts of our infrastructure such as a smart grid.

- *Heterogeneity:* The system contains a number of different, application specific hardware and software components. Abstraction layers and generic interfaces facilitate the implementation for developers and suppliers. By employing middleware solutions, software components can be easily adapted to changes in the hardware components.
- *Large scale:* Current automotive E/E architectures contain several thousands of software components and up to 80 ECUs. Those numbers are expected to increase in the future with the introduction of piloted and highly autonomous driving features. A car that is connected to its surrounding vehicles and the infrastructure forms an even larger system. New challenges with respect to reliability, privacy and adaptivity are expected to emerge with the introduction of such a "cloud architecture". CPS are often very large and complex systems which leads to new challenges with respect to data collection, storage and interpretation.

8 Transfer to other CPS Domains and Related Work

The heterogeneity of domains, in which CPS can play an important role, is growing and numerous. In fact, there is an increasing use of CPS in automotive, chemistry processes, medical instruments, energy, aeronautics, and so on. In each of this domain the employed CPS have different characteristics, for instance the dimension of the systems, their interconnections, the availability, the safety, and reliability, and the different conditions of the environment. In a such scenario categorize and summarize common aspects of CPS in different domains can be difficult. Another complex issue is that usually the design of CPS involves experts from many different disciplines. There may be mechanical engineers, computer scientists, electric and electronic engineers, which manage respectively mechanic, software and e/e parts of the systems. Since the role and is integration in CPS is increasing, also the usual development techniques within software engineering assume an increasing importance and arise new challenges.

We presented in the Section 5 reference architectures in the domain of smart grid and in the Section 6 for the automotive domain. We analyse aspects considering also other domains and classify it in comparison with other modelling frameworks or methodologies. Also the interdisciplinary nature of CPS is part of this transfer and classification process.

8.1 Analysis of the Framework Characteristics

Let's consider some characteristics and elements of the artefacts described in the reference framework. If we consider the flow of energy between the subcomponents, also in almost other domains where the CPS are used, one needs to model it. It is so a founding element for the supply of energy to the involved systems, therefore in the consumption of energy, but it may be also important for the production of energy. As for instance in an electro car, its brake system produces electricity, and so energy, when the brakes are activated and is stored for the motor. Another example may be the energy produced by the movement of a watch, in order to recharge its batteries, we have again a produce/consume energy CPS. Therefore, the modelling of the energy flows, storages, consumption and production units is in each domain important.

Another important characteristic, presented in our framework, is the subdivision of the modelled system into subcomponents and units. This concept is known with different names, as for instance modularity, with which a system is subdivided into logical or functional interconnected elements/units, in order to reduce the complexity of the visualization, produce a representation that covers the final hardware/software architecture and subdivides the system between the involved disciplines. The presented framework provides also a structured and graphical representation, where the flow of the date and energy may be furnished with additional information.

8.2 Related Work

Mechatronic UML is an approach for the interdisciplinary modelling especially of the software part of mechatronic system [6]. It is based on the model driven development, supports the definition of links between engineering disciplines and refines a subset of UML. The tool supports the check of safety and liveness properties through formal verification and code generation from the models. The architectural view-model implemented in 3+1 SysML models disciplines involved in the development of CPS in three views, that are mechanic, electronic and software [61]. For the description of the models is employed SysML and to analyse and simulate them can be utilised discipline specific tools. Mechatronic Modeller is a software tool for multi-disciplinary systems, based on the concept of reuse of solution patterns [2].

Modelica is an object oriented and component based language, for the modelling and simulation of heterogeneous physical systems [62]. It is used for discrete, continuous and hybrid systems; but it requires a high level of detail that may make difficult handle different level of abstractions. It is based on a standard library and there are numerous tools that support Modelica, as for instance OpenModelica Mosilab and Dymola. Matlab combined with its dynamic simulation toolbox Simulink [11] is a well-known industrial tool for the modelling of mechatronic systems, supporting the multidisciplinarity in a model based environment. Industrial Engineering tools that support all the design phases of automation mechatronic systems are SIMATIC automation designer [12], COMOS [13], and Eplan Engineering Center [14].

In the first viewpoint there are the requirements artefacts of the system, that may be in formal or informal, usually text files but they may be also graphical. The second viewpoint is the functional view, obtained from the requirements, where are represented the functionalities of the system, in an hierarchical way; also not functional requirements are included. From the functional view the logical view is derived, the system is decomposed in subcomponents, the interface is well defined and it is possible to simulate the system through the message/signal exchanged between the components. The last view is the technical, which contains the implemented hardware and the final implementation of the system. The idea is to make another step from the logical to the technical view, anyway in the logical was possible to execute,

---

[11] http://www.mathworks.com
[12] http://www.siemens.com/simatic
[13] http://www.siemens.com/comos
[14] http://www.eplan.de

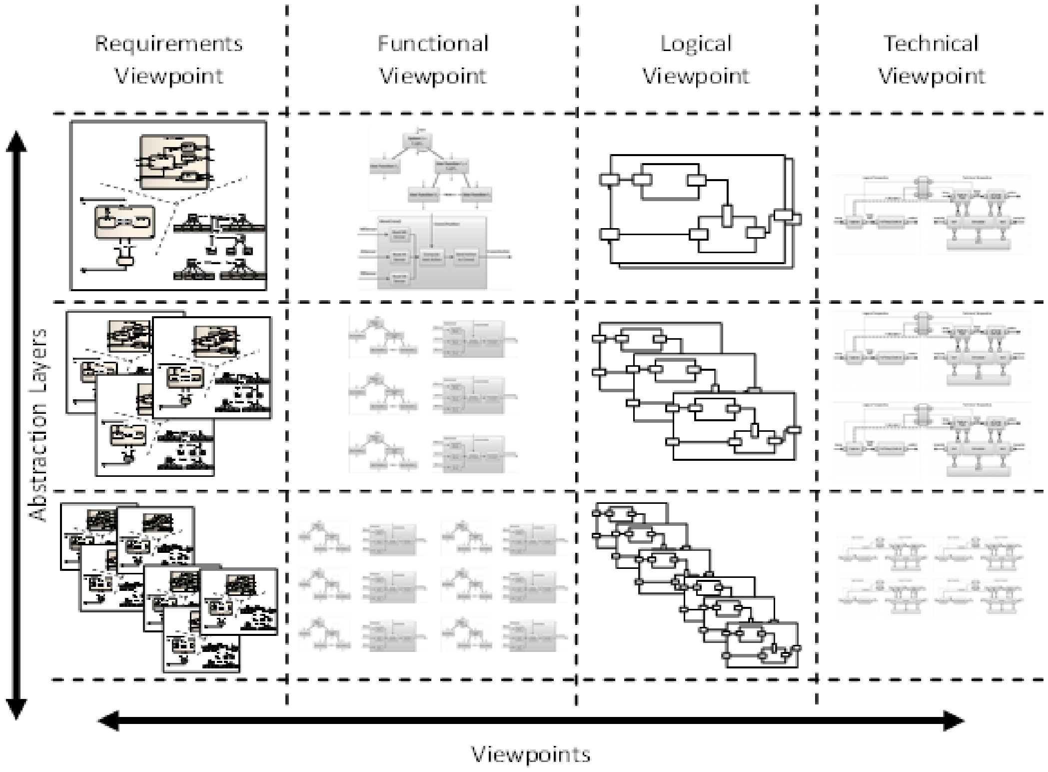


Fig. 13. The SPES matrix: level of granularity and software development view

simulate and also verify the system.

Interdisciplinary Modelling

The different expertises involved in the design of CPS can be summarized in software, mechanical and electro technical engineering. In the last years the role played by the software part, in the control and automation of CPS is certainly increasing. One direction in the research is to bring the already consolidated software engineering fundamentals, in order to include also the support for other disciplines artefacts. Considering that the final hardware specifications require discipline specific models, a control based logical representation of the system can be obtained from the software models. This representation can provide important features as simulation and verification. Ideally these artefacts would be in logical level of the SPES matrix. In order to obtain this result it is necessary to evolve the software models to interdisciplinary specifications.

FOCUS provides a functional hierarchy and a logical architecture for distributed and embedded systems [9]. These models are more suitable to describe software components, specific aspects of other disciplines are not yet covered. A crucial aspect of FOCUS is that it provides component communication through i/o stream functions, with data types, between the components.

An output stream of a component can be the input stream of a not necessarily different component. Besides this kind of interfaces it is possible to define an hierarchy of components, defining components inside other components. The component at the lower levels must contain an implementation, which is usually a functional specification or a state machine. The components read the input and produce the output with a discrete time step. This might limit the support for other disciplines in FOCUS. In fact, components as actuators and sensors have a deep integration with the environment and have to be ready to react to the physical world. In the models will be so important to support continuous data and time. A well-known paradigm for the representation of continuous and discrete time systems is the hybrid automaton introduced by Henzinger [27]. Hybrid automata enable continuous evolutions through differential equations associated to variables, and discrete transitions between the states of the automaton. An extension of FOCUS to support continuous i/o streams is introduced in [10]. Therefore, real-time system requirements can be covered by continuous time i/o streams. In this paper are defined the characteristics of the i/o streams, which are elaborated by the FOCUS components, considering the behaviour at the interface level. An internal logical implementation for FOCUS components was presented in [12], which is based on a modified version of i/o hybrid automata. In this work, another topic about hybrid systems was considered, that is the simulation. A full continuous simulation in a supporting tool would be too complex, therefore are discussed some approaches to discretise the continuous signal produced in the components. These approaches are based on sampling, that is, at selected times and stepwise the values for the continuous variables are calculated. The interval between two elaborated values is said period and remains usually constant. In [12] the period varies in order to improve the precision of the simulation and it is said dynamic sampling.

In an ideally higher level of abstraction, respect to the modelling theory FOCUS, there is the interdisciplinary approach elaborated in the research project MODEMAS[15]. Anyway, the system architecture presented in this project, is still founded over FOCUS, but has discipline specific characteristics. In an hierarchical way and with different level of abstraction, is it possible to define disciple specific components that are interconnected in the same architecture. The interface is still modelled with streams, but they still have a discipline identity, that is, the software part that provides data streams; the mechanic part material streams; and the electric part energy streams.

---

[15] http://www.dfg-spp1593.de/index.php?id=39

8.3 CPS Domains

We analysed in detail the role of CPS in energy and automotive domains, but the application of CPS is not limited to them. Other domains are chemistry, aerospace, medical and mechanical. In Section 5 we mentioned the fundamental smart grid activities, most of them describes needs and challenges of other CPS domains. Considering the presented reference architecture IEEE P2030, composed of three different subdomains, we could find analogies using these subdomains for other types of CPS. The smart energy subdomain provides the core functionality of a smart grid, it might be possible to transfer this subdomain, as the main functional architecture, to other CPS domains. For instance, it might describe the material production in a mechanical system or the resulting substances in a chemical process unit. The second subdomain is the smart information, regarding date metering, monitoring and management; it might also be a reference description in order to follow or monitor the implementation of general system requirements, not necessarily in the energy domain. The last subdomain is the smart communication system necessary to model the communication within a smart grid and their components. Again, it might be an interface description in different fields, where there is a structure in subcomponent of the system with a defined interface. Moreover, in the IEEE P2030 architecture each perspective is then composed of seven domains, that represent a further level of abstraction in the architecture, where each domain represent a different aspect of the smart grid. Despite these aspects are specific of smart grids, they might be redefined for different application domains, using the same architecture. The same generalization might be also applicable to the referred SGAM reference architecture. The five abstractions views, combined with the domains and the customer premises constitute a 3D structure, where the layers are already general for other domains meanwhile the other two are smart grid specific. A similar approach but domain independent for embedded systems, is presented in the previous section. It is structured in different system views in the so called SPES matrix. Since it is conceived for embedded systems, it has not the same detailed informative level as the SGAM architecture, therefore the SPES matrix might integrate some aspects of the SGAM architecture in order to extend it to CPS.

We analysed the automotive domain in Section 6 presenting the AUTOSAR standard. In this standard are defined five different functional domains, where the subdivision is based on main functional categories. This representation of the system is based on the functionalities and is one standard system description used by model-based engineering methods, for instance in the already mentioned SPES matrix there is a functional view, usually directly derived from the requirements. Anyway, there is some information in the AUTOSAR standard that might be transferred to other domains. The decision to separate in different domain safety and non-safety functionalities might be surely convenient because many CPS work in a safety critical scenario. We referred to the explosion of the complexity of CPS in automotive, that is already a common problem in some domains or it might be in the near future. The main features to overcome this problem, mentioned for the ”Smart Car”, are general characteristics that might be well applied to CPS of other domains.

9 Reference Framework for CPS

The reference engineering framework for CPS represents elements that are useful for the future development of CPS. It should compensate problems and risks we explained in the last sections and act as a reference for developers of several CPS domains.

With respect to the diversity of CPS domains, we see two important elements in the reference framework: Reference architectures and reference workflows. Reference architectures ensure the collaboration and exchange between several CPS domains while reference workflow sketch important activities throughout the development of CPS. Both taken together form the reference framework for CPS and show how the CPS engineering community could tackle the challenges existing in future development.

9.1 Reference Architecture

A commonly used artefact to document the architectural properties of a class of systems is a reference architecture.

We see in the development of a reference architecture an important puzzle piece for the future development of CPS. Therefore we explain what reference architectures are and which role they play in the reference framework.

9.1.1 Reference Architectures in a nutshell

Reference architectures as an own field of research has been discovered in the last years. Therefore only a small number of publications are available that deal with the research on reference architectures.

Reference architectures were already suggested for a plethora of classes of systems such as web servers [26], P2P overlay networks [1], situated multi-agent systems [65] or space data systems [56], just to name a few.

Purpose and nature

A reference architecture can be defined as reference model that captures architectural knowledge about a class of systems; It encompasses common architectural patterns of the class and their interrelation. It must be supported by a unified, explicit, unambiguous, and widely understood domain terminology. [32]

The term *reference architecture* is synonymously used to the term *product line architectures* by some authors, but there are some important differences between both concepts [42]: While reference architectures deal with the range of knowledge in an application domain and are generally on a higher level of abstraction, product line architectures are more specialized on a specific subset of software systems and are concerned with variabilities among products.

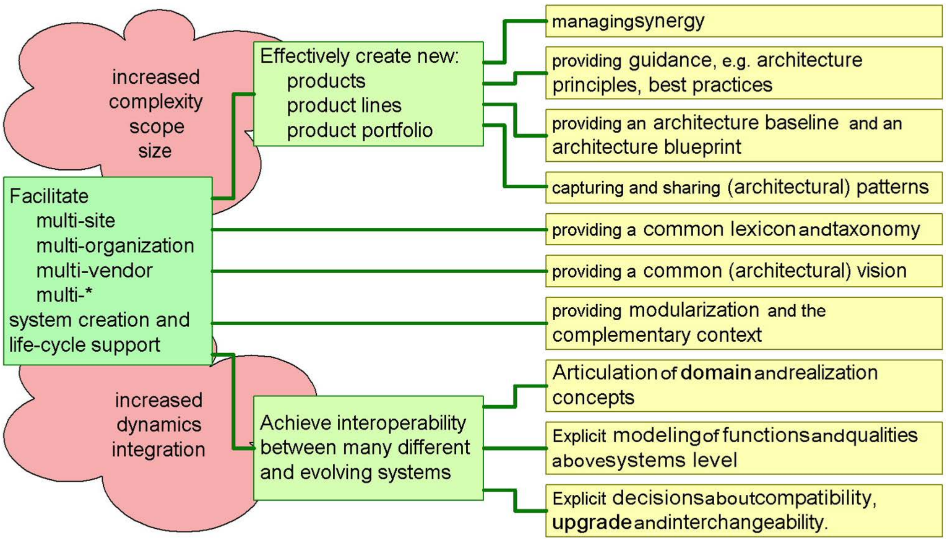


Fig. 14. The objectives of reference architectures, suggested by Cloutier et al. [15].

Nevertheless can reference architectures act as a basis for the development of product line architectures.

The so called "multi-effect" leads to the development of reference architectures. When an application domain is faced with

- increasing complexity, scope, and size of the system of interest, its context, and the organizations creating the system and
- increasing dynamics and integration: shorter time to market, more interoperability, rapid changes, and adaptations in the field

in a critical mass, reference architecture were developed to tackle these problems [15].

According to Cloutier et al. [15], reference architectures provide :

- a common lexicon and taxonomy
- a common (architectural) vision
- modularization and the complementary context

Reference architectures are useful for

- managing synergy
- providing guidance, e.g., architecture principles, best practices
- providing an architecture baseline and an architecture blueprint
- capturing and sharing (architectural) patterns.

on the basis of proven concepts within an application domain (Figure 14).

Content elements and first models

A tremendous amount of knowledge about a domain and their systems is available, both implicit as well as explicit in design repositories and documentation. However, this knowledge is not structured and captured in ways that allow the exploration of the accumulated concepts and solutions.

Therefore a reference architecture has to explicitly capture the domain-specific, relevant but generalized knowledge. It has to make it accessible in digestible proportions, without losing too many essential detail. The right level of abstraction and the flexible access to the content of a reference architecture are crucial for its success.

Many authors therefore suggest to use knowledge models such as ontologies as a central concept for the creation of reference architectures [15,32,43]. Nakagawa et al. suggest RAModel, a reference model for reference architectures (Figure 15) which includes the main elements that could be discovered in a literature study of published reference architectures.

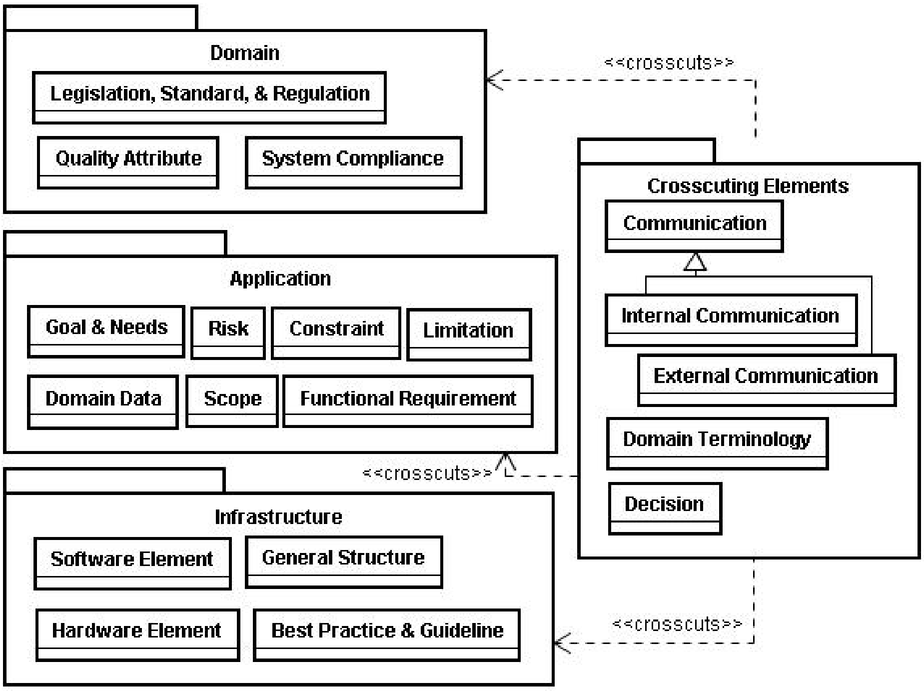


Fig. 15. RAModel, a reference model for reference architectures, suggested by Nakagawa et al. [43].

RA Model shows that reference architectures capture a broad knowledge existing in application domains:

- Domain legislations, standards, and certification processes, which impact systems in that domain are represented in the **domain** cluster
- elements that provide a good understanding about the reference architecture, its capabilities and limitations, elements related to the business rules

(or functionalities) that could be present in software systems built from the reference architecture are represented in the **application** cluster

- elements that could be used to build the software systems based on the reference architecture to enable systems to automate, for instance, processes, activities, and tasks of a given domain are represented by the **infrastructure** cluster
- set of elements that are usually spread across and/or tangled with elements of other three groups (domain, application, and infrastructure). are represented by the **crosscutting** cluster

Development and structure

The question of how to develop or maintain reference architectures is still an open issue [41]. Many authors demand to base the reference architecture on an empirical foundation of existing and proven system concepts and the best practices found in development [15,21,32,48].

A bottom-up approach and a three-layered model were suggested by Irlbeck et al. [32] to develop reference architectures that are built on an empirical basis:

- The **system level** incorporates knowledge gained through the analysis of individual systems in a domain. It represents the empirical basis for the creation of a reference architecture.
- The **reference level** represents the reference concepts, the main elements of the reference architecture. Through relational linking of reference concepts to the system concepts of the system level a tracing to individual system concepts can be established.
- The **query level** offers an uniform interface to the reference architecture in the form of queries. The query level addresses stakeholder specific needs, eases the access to the reference level and increases the tangibility of the reference architecture.

The three levels represent a first rough structure for the creation and usage of a reference architectures. Furthermore a 7 step procedure for the creation of reference architecture on the basis of the three-layer model is proposed by the authors [32], which allows to structure the process.

9.1.2 The Role of Reference Architectures in the Reference Framework

We suggest to discuss the creation of reference architectures for several CPS domains. In the last section we presented the current research on reference architectures. As we showed, first models for the content of reference architectures (RAModel by Nakagawa et al. [43]) as well a structure for its development and usage (Bottom-up approach by Irlbeck et al. [32]) are already focused by current research.

Although there are many open challenges, the role of reference architectures will be important for future CPS engineering. The presented results show that reference architectures can represent a homogeneous description mechanism for heterogeneous systems, as explained in sec. 3 and are a way to solve the challenges of CPS, as exemplified by sec. 4.

Therefore, reference architectures for CPS domains of a system under development, representing important CPS domain knowledge are included in the reference engineering framework.

Some important elements which we identified while analysing the CPS domains of Smart Grid in sec. 5 and Automotive in sec. 6 are not represented yet reference architectures, but both domains show that there is potential to establish reference architectures that capture the commonalities we identified in 7.1.2.

The task that lies ahead is the development of reference architectures for CPS domains that are built on an empirical basis and that use a standardized, accessible model to represent the elements of reference and their relation. These reference architectures can be included in the reference engineering framework to efficiently develop future CPS, that are aware of other CPS domains and their important contexts.

9.2 Reference Workflow

As each CPS has unique domain specific requirements with respect to the development workflow, it is not feasible to propose a generic workflow that can be applied to all systems. Instead, we extracted general system aspects that have to be covered by a CPS development workflow derived from the preceding sections.

(i) **Specification of the application context:** The crucial property of a CPS is the tight integration into its physical, organizational and social application context. Legacy systems can also part of the application context. Discovering and modelling the important properties of the application context, which is commonly referred to as domain engineering, should be the initial step.

(ii) **Specification of features and functionality:** The behaviour of the system has to be defined covering functional and quality aspects. We propose a model based approach as these models can be seamlessly used during specification, implementation, testing, integration and maintenance of the system.

(iii) **Specification of technical components:** The technical building blocks of the technical architecture, e.g. types of sensors, actors, aggregates and other physical/technical elements such as communication networks have to be defined and specified accordingly. A model based approach can support the development of a technical architecture as it supports the evaluation of several alternatives based on different combination of the architecture's building blocks.

(iv) **Evaluation of the logical architecture:** As logical and technical architecture are developed in parallel, it is essential to analyse the consistency of the developed and assembled architecture. Especially for multifunctional systems, the behavioural integrity needs to be preserved as the feature has to perform as expected in conjunction with a number of other features needed to fulfil the system's overall objective.

(v) **Deployment definition:** Definition of an initial deployment for the base system. For fully runtime-adaptive systems, it is sufficient to initialize the system with the application context and the definition of the expected set of features and functionality.

9.3 Future Refinement of the Framework

The above sketched Reference Framework was contoured with the goal of tackling the challenges posed by the engineering of the envisioned CPS. This framework is still embryonic, i.e., in a rudimentary stage and with potential for further refinement. Real-world developments only can contribute to its ripening. The considerations that guided the proposal of the previous section can assuredly contribute to those case studies. Because of this reason, we propose below a checklist that will certainly serve as a guide. The checklist below contains several relevant dimensions that must be taken into account for the design, realization and maintenance of CPS.

Human socio-technical dimension:

(1) what is responsibility of the system?

(2) what is responsibility of the human operator?

(3) what are the boundaries of (shared) control?

(4) what to do in case of conflict between behaviour demanded by human and pertinent reaction as inferred by the system?, are there any regulations to be taken into account?

(5) what must be highlighted (in instructions, online help, etc.) in order to avoid misbehaviour?

Technical realization:

(6) what data/behaviour is competence of the system under development?

(7) how is data to be managed, accessed, protected?

(8) how are data and intelligence to be distributed?

(9) what is stored/decided locally?, what in the cloud?

(10) how to guarantee high availability of (large volume of) data?, what to do in case of unavailability?

(11) how to measure and guarantee quality of service?

Data dimension:

(12) what is data to be collected by the system (as measured by sensors, among others)?

(13) what is the query frequency?, are there any trigger mechanisms besides query frecuency?

(14) data replication: which are push mechanisms?, which ones pull mechanisms?

(15) how to storage data?, how to aggregate data?, in particular: preparation and processing, data mining, durability, substitute values, plausibility checking, precision, data dependencies

Technical realization of communication:

(16) what data/behaviour is not competence of the system under development?, and how to access data and trigger behaviour of other systems?

(17) what should be communicated via dedicated networks?, what can be communicated using public networks?

(18) how to realize channels?, e.g., glass fiber, wireless, powerline, etc.

(19) what bandwidth is at least necessary?

(20) how to measure and guarantee quality of communication?, in particular, how to ensure security against eavesdropping?

(21) how to guarantee unlinkability, in particular in combination with surrounding systems, and anonymisation?

(22) how to meansure availability and/or performance?, and if necessary, how to resort to other communication means?

(23) what property rights must be observed?, what liability is implied?

Safety and security:

(24) choice of authentication mechanisms

(25) authorization procedure and its administration and maintenance

(26) non-repudiation, i.e., proof of integrity and origin/genuineness of data

(27) validation and verification of the system, certification

(28) IT security including information privacy, encryption, firewalls and other intrusion detection mechanisms as well as protection from unplanned events and natural disasters; security controls with respect to: confidentiality, integrity, availability, accountability and assurance services

Criticality:

(29) data reliability (by, e.g., data redundancy together with decisions procedures for avoidance of anomalies and corruption)

(30) degree of fault tolerance, self healing mechanisms, evolvability

(31) black start capability

Architecture:

(32) how are the different systems parts (e.g., components) to be structured?

(33) what are the key functions, so that they are, e.g., easy accesible and maintainable?

(34) how to organize an effective and efficient data flow?

(35) how are the interfaces to be specified and put at disposal of other CPS?

(36) how to organize a stakeholder-specific view, on the one hand, and a application-specific views, on the other, such that the associated goals (i.e.,complexity reduction, abstraction, distraction reduction, access protection, etc.) are achieved while consistency is preserved?

(37) how to document rationales, so that evolution is facilitated?

(38) how to link cross reference domains?

(39) how are functional logical, and technical levels defined and documented?

Physical World:

(40) choice of sensors according to relevant criteria as, e.g., reliability, replication, smartness

(41) choice of actuators according to durability, weather conditions, etc.

(42) repercussion on physical systems, in particular accident avoidance

(43) sampling, frequency (and, related to this, discrete vs. continuous modelling)

(44) how to correctly assess spatial/temporal aspects?

Legal and social dimension:

(45) laws, norms, regulations, by-laws and their impact on system design

(46) ease of use, intuitiveness

(47) user acceptance (by means of, e.g., living labs and mockups)

(48) how to eliminate/reduce the impact of dropouts (e.g., non-networked cars)

Market/business view:

(49) what is the market structure, how does the system under development fits?

(50) what is the role of the system in its context of placement?

(51) what are the business objectives, and how to reach them?, including goals, time to market, life cycle, etc.

(52) what are the business processes for the system under developement as well as for the systems in the environment?, e.g., how to reconcile collaborating systems of different life cycle length

Terminological view:

(53) denomination of system components, homonymy avoidance

(54) cross-domain communication: glossary, ontologies, specification of interfaces, etc.

(55) conceptualization of the system and its relevant functionality, in such a way that its marketing as well as its interconnection with systems and users from other domains remains simple and easy

10 Impact

In this section we discuss the different kinds of impacts of a reference framework and highlight its benefit for the development of CPS.

Impact 1: Representation of CPS domain knowledge in a standardized way for the use in development of CPS

CPS domains have to share knowledge in order to collaboratively work on new systems that span across multiple domains. This knowledge should not be elaborated for each system separately because of efficiency, reuse and standardization issues. The reference engineering framework for CPS proposes to develop reference architectures for many CPS domains. Whenever a new system is build, its developers can consider the needed domain representation and structure the development of the system. For the creation of product lines this representation has an impact on the development as well. A possible instantiation scheme is presented by Cloutier et al. [15] and shown in Figure 16. The figure shows as well how feedback from the development of CPS can be used as a feedback for the further development of reference architectures. With this mechanism applied, reference architectures are kept actual and can be used for the further development of CPS.

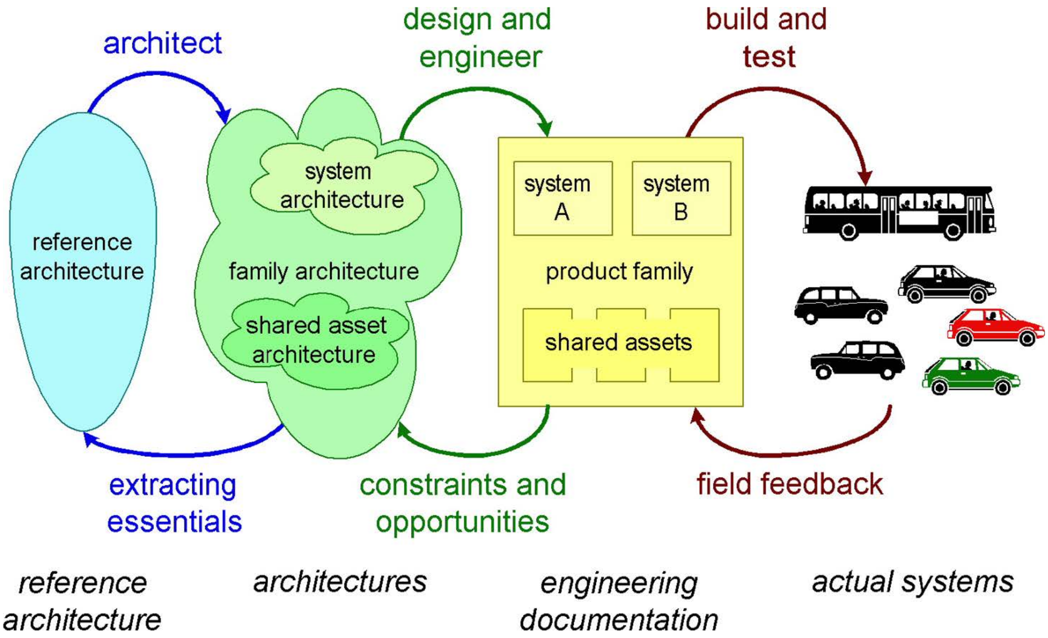


Fig. 16. The development of individual CPS or product lines is enabled through the instantiation scheme of Cloutier et al. [15]

Impact 2: Enabling collaboration between CPS domains through a accessible, standardized, shared and active terminology

Systems do not consist only out of building blocks and their connection. They involve an often implicitly defined terminology. Humans need terms to discuss about systems. Terms identify abstract elements that are part of a system (such as components, functions, services, data types) or which characterize an abstract system concept. Terms also refer to parts of a system's context, domain concepts, roles or other important entities. Commonly used terms often lack an explicit definition of their meaning, their context of definition, existing abbreviations, language and the definition of linguistic relations to other terms (e.g. synonymy, hyponomy, hypernymy, meronymy or indication of false friends, translation in other languages).

Different systems may share terms, especially when they belong to the same class of systems. Nevertheless the meaning of these terms in the context of a class of systems is ambiguous and not defined, as there is often no frame of reference.

To this end our for reference frameworks addresses terminological aspects by reference architectures, describing terms in use in context of systems and on a reference level and their respective meaning. They provide mechanisms to define terms, to interrelate them, to link them to architectural concepts and to express linguistic characteristics of e.g. synonymy.

The activity of establishing an actively used, clear terminology for CPS is embedded in the reference architecture of the presented reference framework. The identification of terminological aspects of systems engineering enables different stakeholders over different domains and disciplines to work collaboratively without the risk of misunderstandings.

Impact 3: Transdisciplinary Collaboration

As outlined in section 8 there are many common ways to model a system's structure and behaviour. Unfortunately these different modelling and description techniques complicate the collaborative interaction between different CPS domains.

A uniform representation can avoid this problem by representing concepts of different systems in the same way. This representation permits the uniform description of relevant concepts on the reference and system level and the mapping between both levels.

To cover a wide variety of descriptions the reference architecture within the reference framework defines a set of basic concepts (e.g. components, functions, data structures or interfaces) and allows their flexible connection to model certain CPS.

Within the workflow the reference framework addresses

Impact 4: Clear definition of abstraction layers and viewpoints

Many system documentations differ in their levels of detail, completeness and preciseness. For some tasks detailed descriptions offer too many details, while high level documentations as well as fragmentary documentations may be too vague. Standardized abstraction layers and a definition of common viewpoints throughout CPS development allow to create a common perspective on the system under development over many disciplines being involved.

The reference framework should define clear levels of abstractions, based on the structure presented in the SPES matrix (cf. Figure 13) and the presented workflow.

As the SPES matrix focuses on the development perspective of embedded systems, other CPS domains have to be presented in own abstractions and viewpoints which allow an effective way to reduce complexity as well as apply refinement to include new details. It must be capable to address abstract views in early stages of development which can be refined in course of the development of the framework.

Impact 5: Fusion of open and rigorously developed systems

In the future two types of systems will collide: Traditional enterprise systems on the one hand and embedded systems on the other hand. The properties of these two systems differ a lot. While enterprise systems often use legacy systems or code and have to maintain a huge set of functionality for a long time, embedded systems are fulfilling a dedicated function within a larger mechanical or electrical system and are constantly replaced.

The fusion of both sectors in future development for CPS requires an evolution of both worlds: While enterprise systems have to be opened and standardized while maintaining their service, embedded systems have to be connected to a larger context that rip them out of their dedicated context.

Large CPS as the smart grid show, that this fusion is complicated and cost-intensive. The efficient evolution of both worlds is a huge impact, the proposed reference framework could offer.

Impact 6: Dynamic and spontaneous interaction of CPS

The further connection of CPS and their dynamic interaction challenge the further development of new paradigms. As we have shown in Section 3, the identification and description of different system contexts are of paramount importance for the possible interaction of different CPS. The change of location of some CPS (e.g. in the automotive or transport domain), the mobile interaction with CPS (e.g. through mobile communication) or infrastructural CPS (e.g. electrical power grid) that span across multiple countries imply a dynamic context for some CPS.

Spontaneous and dynamic interactions between different CPS will increase in future and should sometimes not be stopped at country borders. The international collaboration on CPS contexts such as the harmonization of legislation, regulation and standards ensure a future quality of service.

Already today there are systems as the mobile communication network that make it possible for customers to travel across countries while the desired service is maintained. The mobile phone can dynamically change its technical context (new transmission towers, mobile providers or transmission bands), adapts to the new context and continues working. National laws ensure the legal framework for this functionality and regulate the fees for the service.

The further investigation of desirable functionality and its (possibly international) implementation while considering different CPS contexts (e.g. technical, legal, social, commercial, ecological) and the connection of different available CPS are tasks that traditional development frameworks cannot support.

Therefore, the reference framework has to be refined to address the future needs for the dynamic interaction of CPS. We address this issue with the use of reference architectures and with a checklist that includes already several important dimensions that have to be considered for the dynamic interoperability of CPS.

11 Conclusion

Characteristic properties of CPS that are, on the one hand, inherent to open systems as, e.g., their portability and interoperability with each other, their heterogeneity and self-evolvability, and, on the other hand, inherited from embedded systems as, e.g., a life cycle consisting of decommission followed by design, building and commission, whose constraints fall in the category of hard real-time, are very difficult to tackle using traditional methods tailored for one or the other realm. Nevertheless, agile and rigorous methods can be combined, yielding a framework useful for the development of CPS.

This possibility has been investigated above to a certain level of detail. A number of lines of actions has been proposed, in order to devise a method suitable for the realization of the novel systems. The reference framework seeked for has to fulfil specific requirements, which have been precisely stated above. Moreover, a Reference Architecture and a Reference Workflow along with a checklist for the engineering of CPS have been exemplarily delineated, that comply with those requirements. This Reference Framework should now be validated by means of a case study of appropriate size. We furthermore expect a discussion be triggered by the statement of this document. The CPS vision can only be achieved by means of a common understanding of the interoperability of these systems, as well as a coordinated approach to their interaction.

References

[1] Aberer, K., L. Alima, A. Ghodsi, S. Girdzijauskas, S. Haridi and M. Hauswirth, *The essence of P2P: A reference architecture for overlay networks*, in: *5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005)*, IEEE, 2005, pp. 11–20.

[2] Anacker, H., R. Dorociak, R. Dumitrescu and J. Gausemeier, *Integrated tool-based approach for the conceptual design of advanced mechatronic systems*, in: *5th Annual IEEE International Systems Conference, Montreal, Quebec, Canada, April 4-7*, 2011.

[3] Anders Troi et. al., *SMART GRID FORSKNINGSNETVÆRKET - Roadmap for forskning, udvikling og demonstration inden for Smart Grid frem mod 2020* (2013).

[4] Andreas Lugmaier et. al., *Roadmap smart grids austria* (2010).
URL http://www.stadt-salzburg.at/pdf/2010_roadmap_smart_grids_austria.pdf

[5] AUTOSAR Group, *AUTomotive Open System ARchitecture (AUTOSAR) Release 4.1* (2013).

[6] Becker, S., C. Brenner, S. Dziwok, T. Gewering, C. Heinzemann, U. Pohlmann, C. Priesterjahn, W. Schäfer, J. Suck, O. Sudmann and M. Tichy, *The MechatronicUML Method - Process, Syntax, and Semantics*, Technical report, Software Engineering Group, Heinz Nixdorf Institute University of Paderborn (2012).

[7] Bernhard, M. et al., *The Software Car: Information and Communication Technology (ICT) as an Engine for the Electromobility of the Future* (2011).

[8] Bouhafs, F., M. Mackay and M. Merabti, *Links to the future*, Power and Energy Magazine, IEEE **10** (2012), pp. 18–23.

[9] Broy, M., *Service-oriented systems engineering: Modeling services and layered architectures*, Lecture Notes in Computer Science (2003), pp. 48–61.

[10] Broy, M., *System Behaviour Models with Discrete and Dense Time*, in: S. Chakraborty and J. Eberspächer, editors, *Advances in Real-Time Systems* (2012), pp. 3–25.

[11] Bundesministerium für Wirtschaft und Technologie (BMWi), *E-Energy: IKT basiertes Energiesystem der Zukunft* (2009).
URL http://www.bmwi.de/BMWi/Redaktion/PDF/Publikationen/Technologie-und-Innovation/e-energy-ikt-basiertes-energiesystem-der-zukunft,property=pdf

[12] Campetelli, A., *Dynamic Sampling for FOCUS Hybrid Components*, in: *3th International Conference on Circuits, System and Simulation (ICCSS 2013)* (2013), pp. 402–406.

[13] *CHROMOSOME in 120 Minutes*, Technical report, fortiss GmbH (2012).
URL http://download.fortiss.org/public/xme/xme-tutorial-0.2.1-win32.pdf

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

[15] Cloutier, R., G. Muller, D. Verma, R. Nilchiani, E. Hole and M. Bone, *The Concept of Reference Architectures*, Systems Engineering **13** (2010), pp. 14–27.

[16] Electricity Networks Strategy Group, *A Smart Grid Routemap* (2010).
URL http://webarchive.nationalarchives.gov.uk/20100919181607/http:/www.ensg.gov.uk/assets/smartgrid_routemap_executive_summary_final.pdf

[17] Electricity Networks Strategy Group, *A Smart Grid Vision* (2010).
URL http://webarchive.nationalarchives.gov.uk/20100919181607/http:/www.ensg.gov.uk/assets/ensg_smart_grid_wg_smart_grid_vision_final_issue_1.pdf

[18] ESO, "The M/490 Mandate Smart Grids Reference Architecture," European Standards Organizations (2012).

[19] Fang, X., S. Misra, G. Xue and D. Yang, *Smart Grid - The New and Improved Power Grid: A Survey*, IEEE Communications Surveys and Tutorials **14** (2012), pp. 944–980.

[20] FutuRed, *Documento de Vision Estrategica 2030* (2011).
URL http://www.futured.es/wp-content/uploads/downloads/2012/06/Documento-de-Vision-Estrategica-2030.pdf

[21] Galster, M. and P. Avgeriou, *Empirically-grounded reference architectures: a proposal*, in: *Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS*, ACM, 2011, pp. 153–158.

[22] Gamatié, A., S. Le Beux, É. Piel, A. Eiten, R. Ben-Atitallah, P. Marquet and J.-L. Dekeyser, *A Model-Driven Design Framework for High-Performance Embedded Systems*, Technical Report 6614, Institut National de Recherche en Informatique et en Automatique (INRIA) (2008). URL http://hal.inria.fr/docs/00/31/11/15/PDF/RR-6614.pdf

[23] Geisberger, E., 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.

[24] Gharavi, H. and R. Ghafurian, *Smart Grid: The Electric Energy System of the Future*, Proceedings of the IEEE **99** (2011), pp. 917–921.

[25] Giordano, V., A. Meletiou, C. F. Covrig, A. Mengolini, M. Ardelean, G. Fulli, M. S. Jiménez and C. Filiou, *Smart Grid projects in Europe: Lessons learned and current developments*, Joint Research Centre (JRC) report on SG projects in Europe (2013).

[26] Hassan, A. and R. Holt, *A Reference Architecture for Web Servers*, in: *Reverse Engineering, 2000. Proceedings. Seventh Working Conference on*, IEEE, 2000, pp. 150–159.

[27] Henzinger, T. A., *The Theory of Hybrid Automata*, in: *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, LICS ’96 (1996), pp. 278–292.

[28] Heuser, L. and W. Wahlster, editors, “Internet der Dienste,” acatech DISKUTIERT, Springer, Berlin, 2011.
URL http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/acatech_diskutiert/acatech_Diskutiert_Internet-der-Dienste_WEB_02.pdf

[29] Huang, K., 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)* (2012), p. 224.

[30] IEEE, “IEEE P2030 - Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads,” Institute of Electrical and Electronics Engineers (2011).

[31] International Standards Organization, *ISO/DIS 26262 - Road vehicles - Functional safety* (2011).

[32] Irlbeck, M., D. Bytschkow, G. Hackenberg and V. Koutsoumpas, *Towards a Bottom-Up Development of Reference Architectures for Smart Energy Systems*, in: *Software Engineering for the Smart Grid (SE4SG), 2013 International Workshop on*, IEEE, 2013, pp. 9–16.

[33] Jong, H. W. D. and W. G. Shepherd, “Pioneers of Industrial Organization: How the Economics of Competition and Monopoly Took Shape,” Edward Elgar Publishing, 2007.

[34] Kleinknecht, A., *Are there Schumpeterian waves of innovations?*, Cambridge Journal of Economics **14** (1990), pp. 81–92.

[35] Kowert, R. and J. Oldmeadow, *(A)social reputation: Exploring the relationship between online video game involvement and social competence*, Computers in Human Behavior **29** (2013), pp. 1872–1878.

[36] Kutner, L. and C. Olson, “Grand Theft Childhood: The Surprising Truth About Violent Video Games and What Parents Can Do,” Simon & Schuster, 2008.

[37] Ladkin, P., *Report on the Accident to Airbus A320-211 Aircraft in Warsaw on 14 September 1993*, Main Commission Aircraft Accident Investigation Warsaw (1994).

[38] Li, X., X. Yu, A. Wagh and C. Qiao, *Human factors-aware Service Scheduling in Vehicular Cyber-Physical Systems*, in: *30th IEEE International Conference on Computer Communications (INFOCOM’2011)* (2011), pp. 2174–2182.

[39] Loiret, F., R. Rouvoy, L. Seinturier and P. Merle, *Software Engineering of Component-Based Systems-of-Systems: A Reference Framework*, in: *14th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE'11)*, 2011, pp. 61–65.

[40] Marchetti, C., *Society as a Learning System: Discovery, Invention, and Innovation Cycles Revisited*, Technological Forecasting and Social Change **18** (1980), pp. 267–282.

[41] Muller, G. and P. van de Laar, *Researching reference architectures*, in: *Views on Evolvability of Embedded Systems*, Springer, 2011 pp. 107–119.

[42] Nakagawa, E. Y., P. O. Antonino and M. Becker, *Reference architecture and product line architecture: A subtle but critical difference*, in: *Software Architecture*, Springer, 2011 pp. 207–211.

[43] Nakagawa, E. Y., F. Oquendo and M. Becker, *RAModel: A Reference Model for Reference Architectures*, in: *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on*, IEEE, 2012, pp. 297–301.

[44] Nakanishi, H., *Japan's Approaches to Smart Community*, in: *IEEE Smart Grid Comm*, 2010.

[45] National Institute of Standards and Technology, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Version 2.0," (2012).
URL `http://www.nist.gov/smartgrid/upload/NIST_Framework_Release_2-0_corr.pdf`

[46] Olson, C., L. Kutner, D. Warner, J. Almerigi, L. Baer, A. N. II and E. V. Beresin, *Factors Correlated with Violent Video Game Use by Adolescent Boys and Girls*, Journal of Adolescent Health **41** (2007), pp. 77–83.

[47] Rajkumar, R. R., I. Lee, L. Sha and J. Stankovic, *Cyber-Physical Systems: The Next Computing Revolution*, in: S. Sapatnekar, editor, *47th Design Automation Conference (DAC 2010)* (2010), pp. 731–736.

[48] Ramesh, B. and M. Jarke, *Toward Reference Models for Requirements Traceability*, Software Engineering, IEEE Transactions on **27** (2001), pp. 58–93.

[49] Ramos, A. L., J. V. Ferreira and J. Barceló, *Model-Based Systems Engineering: An Emerging Approach for Modern Systems*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (2012), pp. 101–111.

[50] Reenskaug, T., "Working with Objects: The OOram Software Engineering Method," Prentice Hall, 1996, with the collaboration of Per Wold and Odd Arild Lehne.
URL `http://heim.ifi.uio.no/~trygver/1996/book/WorkingWithObjects`

[51] Riehle, D., "Framework Design: A Role Modeling Approach," Ph.D. thesis, ETH Zürich N. 13509 (2000).

[52] Roberts, D. and R. Johnson, *Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks*, 3rd Conference on Pattern Languages and Programming (PLOP 1996) (1996), pp. 471–486.

[53] Rohjans, S., M. Uslar, R. Bleiker, J. González, M. Specht, T. Suding and T. Weidelt, *Survey of Smart Grid Standardization Studies and Recommendations*, in: *Smart Grid Communications*, IEEE, 2010, pp. 583–588.

[54] Schmidt, D., *Guest Editor's Introduction: Model-Driven Engineering*, IEEE Computer **39** (2006), pp. 25–31.

[55] Schmidt, D., K. Balasubramanian, A. Krishna, E. Turkay and A. Gokhale, *Model-driven Development of Component-based Distributed Real-time and Embedded Systems*, in: S. Gérard, J.-P. Babau and J. Champeau, editors, *Model Driven Engineering for Distributed Real-time and Embedded Systems*, Wiley, 2005 .

[56] Shames, P. and T. Yamada, *Reference Architecture for Space Data Systems*, in: *5th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, 2003.

[57] Silverberg, G. and B. Verspagen, *Breaking the waves: a Poisson regression approach to Schumpeterian clustering of basic innovations*, Cambridge Journal of Economics **27** (2003), pp. 671–693.

[58] SMB Smart Grid Strategic Group (SG3), "IEC Smart Grid Standardization Roadmap," (2010). URL http://www.iec.ch/smartgrid/downloads/sg3_roadmap.pdf

[59] Šmihula, D., *Waves of technological innovations and the end of the information revolution*, Journal of Economics and International Finance **2** (2010), pp. 58–67.

[60] State Grid Corporation of China, *SGCC Framework and Roadmap for Strong and Smart Grid Standards* (2010).

[61] Thramboulidis, K., *The 3+1 SysML View-Model in Model Integrated Mechatronics*, IEEE Transactions on Industrial Informatics (2010), pp. 109–118.

[62] Tiller, M., "Introduction to Physical Modeling with Modelica," Kluwer Academic Publishers, Boston, 2001.

[63] Uslar, M., S. Rohjans, R. Bleiker, J. González, M. Specht, T. Suding and T. Weidelt, *Survey of Smart Grid Standardization Studies and Recommendations - Part 2*, in: *Innovative Smart Grid Technologies (ISGT) Europe*, 2010.

[64] VDE/DKE, "The German Roadmap E-Energy/Smart Grids 2.0 (engl. version)," (2013). URL http://www.vde.com/en/dke/std/KoEn/Documents/DKE_Normungsroadmap_ENG-20%20-%20gV.pdf

[65] Weyns, D. and T. Holvoet, *A Reference Architecture for Situated Multiagent Systems*, Environments for Multi-Agent Systems III (2007), pp. 1–40.

[66] Yu, H., M. Spenko and S. Dubowsky, *An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly*, Autonomous Robots **15** (2003), pp. 53–66.