BEYOND ADVANCED MECHATRONICS: NEW DESIGN CHALLENGES OF SOCIAL-CYBER-PHYSICAL SYSTEMS

Imre Horváth

Delft University of Technology

Faculty of Industrial Design Engineering, Delft, the Netherlands.

email: i.horvath@tudelft.nl

Draft paper

Abstract

In the last two decades, an intense shift from advanced mechatronics systems to cyber-physical systems is taking place. The former systems, which integrate mechanical, electronics, computing, control and situated reasoning components, are typically implemented as closed, predefined, controlled, and deterministic systems. The latter systems are characterized by open system boundaries, large functional and structural complexities, self-learning and -reasoning capabilities, partial autonomy, context-driven adaptability, and decentralized decision making. As the latest trend, they are getting more extensively embedded in the fabric of society. First, the episodes of the observable physical, biological, social and technological evolution are overviewed from the aspect of the place and role of information. Then, the distinguishing characteristics of cyber-physical systems are analyzed and the currently on-going transition to social-cyber-physical systems is dealt with. Finally, the paper discusses nine major design challenges raised by the social-cyber-physical system paradigm. As major design challenges: (i) handling aggregative complexity, (ii) static and dynamic compositional synergy, (iii) dynamic and evolutionary operation in time, (iv) multi-abstraction based modeling, (v) system integrity verification and behavior validation, (vi) dynamic scalability towards meta-systems, (vii) transformation of big data, (viii) testable surrogate prototyping, and (ix) robust social compliance are discussed. If we want to address these challenges successfully, then new design principles and system design methodologies need to be developed. The main propositions are that there is an urgent need to intensify multi-disciplinary research in this novel domain of interest, and that new pre-implementation demonstration, prototyping and empirical testing methodologies are also needed.

On the changing role of information in engineered systems

The succession of the major physical, biological, social and technological developments on a historical time scale shows accelerating evolution. Acceleration is obvious if we consider the gradually shortening time periods between subsequent milestones of development. This *accelerating evolution* can also be observed in terms of the emergence and maturation of human-created technologies. As a matter of fact, the shortening of the useful life-cycles in certain technological domains has become so intense that the traditional inception, incubation, maturity, exhausting, and obsolescing pattern of technology evolution can hardly have enough time to happen. This phenomenon is often discussed by science and technology philosophers. However, much less attention is paid to the changing place and role of information in the process of the observable physical, biological, social and technological (PBST) evolution.

As a starting point, let's have a look at the modes of encapsulation and the changing roles of information in the process of PBST evolution. By doing so, we can create a platform for our follow-up discussions. As shown in Figure 1, at the beginning of everything, information basically resided in atomic structures. When genetic materials (such as deoxyribonucleic acid) have evolved, information has been coded, among others, in DNA. When the human brain evolved, information has become embedded also in neural patterns. In the process of formation of human intelligence, capabilities have been developed to externalize and disseminate information by various primary and secondary means of human communication. This was a crucial advancement not only from a cultural point of view, but also from the aspect of aggregating technology-related commonsensical and scientific human knowledge. In the age of industrialization, this aggregation, multiplication and conversion of information to technologies has enabled society-level creation and making, and later on, production of artifacts, systems and processes.

In our modern time, human engineered systems not only encapsulate information and knowledge, but also acquire the potential and abilities to regenerate information, and convert it into operative intelligence. As technology and intelligence continue to integrate, systems with a *high-level working intelligence*, even with a self-reproductive intelligence, can be expected. It can be foreseen that already in the near future, but surely in the farther future, it becomes possible for intelligence to reside and evolve in multi-scale engineered systems. This is assumed to facilitate human presence in and even saturation of a nearby part of the universe.

What is happening in our days is a kind of unrestricted integration of human acquired and artificially generated information with human created artifacts. This is supported by

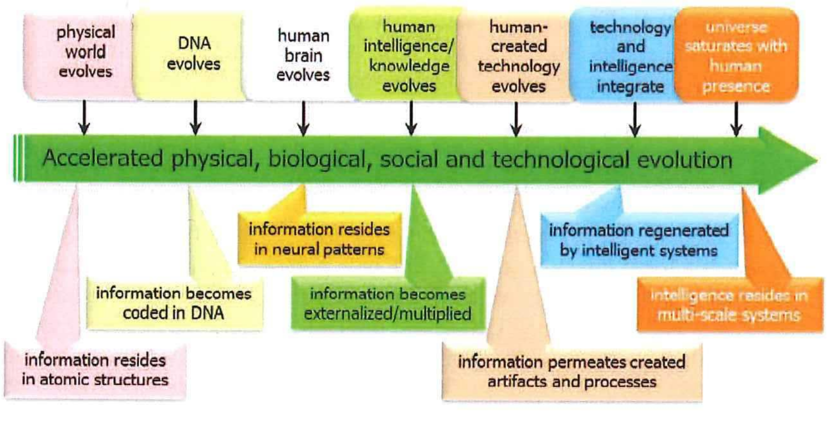

Figure 1 The changing position/role of information in the PBST evolution

the fast development of digital computing and communication technologies which together form one of the current major assets of human society. We have been witnessing the emergence and consolidation of four computing paradigms in the last sixty years. The history of computing commenced with paradigm of mainframe computing. Not more than three decades later, this has been made obsolete by that of networked personal computing. The latter has started its growing into the paradigm of embedded and portable ubiquitous computing two decades later. Though this latter paradigm is still far from being fully exploited or exhausted, the new paradigm of cyber-physical computing is already with us and rapidly evolving. Actually, the first results are already out from the research and development laboratories and getting acceptance in the daily practice. With the advent of cyber-physical systems, the blending of information and knowledge with physical artifacts has reached a very high level, which is referred to as *synergetic integration*. It has to be noted that the emergence of the next possible information processing paradigms such as quantum computing and biological computing, has also started and is advancing with a large pace. The footprints of these paradigms are becoming bigger in scientific research and technology development, though they are still in a premature stage. Experts forecast that they will have a never-before-experienced impact on generating and handling information, in particular by artificial systems. They will permeate and saturate our natural and created environments with qubits-based computing and communication capacity and blend information with artifacts and artificial systems intrinsically.

The reasoning behind structuring of the rest of the paper is as follows: In order to expose the main objectives of complex application systems, first a concise overview of the chronological and conceptual developments is given in the next section. The third section overview the principal characteristics of cyber-physical systems, focusing on high-end implementations, rather than on low-end ones. The fourth section discusses the social and cognitive aspects of cyber-physical systems. The fifth section elaborates on the major challenges at designing cyber-physical systems. Finally, some conclusions are offered and future research work is stimulated.

Illuminating the way to cyber-physical systems

As analyzed by Isermann, the engineered systems were purely mechanical before the second industrial revolution, which was featured by the exploitation of electromagnetism in various forms. This gave floor to the emergence of mechanical systems with electromechanical drives. The next phase of development, at the beginning of the 1930s, witnessed the appearance of electromechanical systems with analogue control. The third technological revolution that was driven by the new digital control and computing technologies in the 1950s made it possible to include digital processors and computers in the control of electromechanical systems. The motion towards incorporating digital computing commenced with electronic control at the beginning of 1970s and was remarkably accelerated by the introduction of the microprocessor in the early 1980s. Actually, this lent itself to the formation of the *discipline of mechatronics*. It was jointly stimulated by the affordances offered by combinations of mechanical, electronic, control and computational technologies, and the growing societal need for more sophisticated industrial systems and infrastructural solutions. Interestingly, in the late 1970s, the Japan Society for the Promotion of Machine Industry (JSPMI) classified mechatronics products into four categories: (i): Class I: primarily mechanical products with electronics incorporated to enhance functionality, (ii) Class II: traditional mechanical systems with significantly updated internal devices incorporating electronics (iii) Class III: systems that retain the functionality of the traditional mechanical system, but the internal mechanisms are replaced by electronics, and (iv) Class IV: products designed with

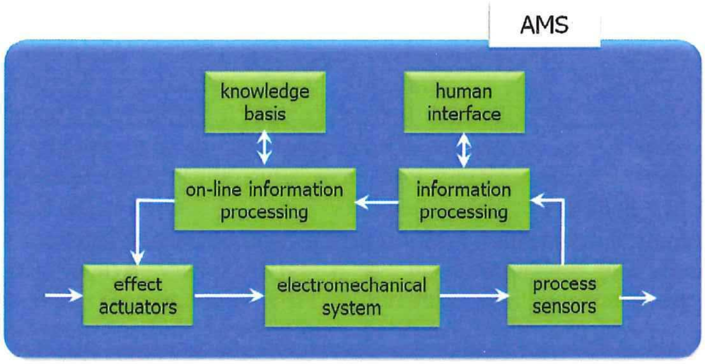

Figure 2 Generic architecture of advanced mechatronics systems

mechanical and electronic technologies through a form of integration that allows enhancing the effectiveness of each other.

Over the last three decades, the above classification has become obsolete in particular due to the recent developments of digital computing and communication. In this time period, mechatronics systems and products have gone through a kind of metamorphosis. *Advanced mechatronics* products such as humanoid robots and service equipment, have been equipped with sophisticated sensors, interfaces, processors, actuators, as well as with complex control algorithms, software agents, and communication means. Exploitation of these resources and knowledge-intensiveness has been the objective of advanced mechatronics since the mid-1980s, with the intent to achieve high level of flexibility and adaptability based on the functionality of the control software. As a result, products are showing a much higher level of functional integration and implementation complexity. The generic architecture of advanced mechatronics systems is shown in Figure 2. Enablers of the development in this direction were not only advanced software design and programming technologies and tools, but also new software architecting concepts such as agent- and component-based implementation.

The affordances of the above technologies and the increased expectations for complex functions and sophisticated structures gave birth to *embedded systems* (ESs). The main objectives of ESs research and development have been to develop functionally smart, structurally adaptive, partially autonomous, and reprogrammable systems. In ESs, computers (more precisely, embedded microprocessors and software means) are used as components to implement these specific functions. While in the case of traditional (totally hardwired) electronic feedback systems physical processes were controlled by the computational elements based upon local and remote computational models and algorithms, in the case of embedded systems, physical processes are monitored and optimized by the computational elements based on sensor information. The traditional feedback-based control systems were designed as closed systems, without operational interfaces. The research in embedded systems largely contributed to moving from closed boundary systems with limited scalability, through cross boundary systems, to fully-scalable open systems. Embedded systems are typically pre-programmed to do specific functions, require real-time behavior, but also constrained in terms of certain resources (e.g. battery-operated). Incorporation of programmable processors in circuits makes the design more robust and thus reduces the design time cycle.

Enabled by digital computing and control, another branch of system development has been *real-time systems* (RTSs). This family of systems has its legacy for the reason that in certain information-intensive engineering systems such as robots, vehicles and medical equipment, it is important not only to provide right output, but also to compute it fast at the right time. Actually, correctness of the control data is a function of the time of delivery (though consistency of the results may be more important than the raw computing speed). Centralized RTSs require real time operating systems (RTOS). One of the most popular one in use today is QNX, which uses a micro kernel for implementing basic system calls, but system level functions such as device drivers, are not part of it. RTSs are either (i) transformational systems (T-RTSs), which take input from the environment at a given time, transform these inputs, and terminate giving the outputs, or (ii) reactive systems (R-RTSs) that have continuous interaction with their environment. While the reaction of R-RTSs on regular (periodic) events can be statically scheduled, random (aperiodic) events must be dynamically recognized, or statistically predicted, when possible.

It has been realized that centralized systems are unsuitable for large-scale system integration because of their (i) high reliance on centralized communication, (ii) high complexity, (iii) lack of scalability, and (iv) the high cost of integration. The use of distributed intelligence technologies avoids these weaknesses. *Distributed intelligence systems* (DISs) are usually based on physical and software agents that (i) operate autonomously, (ii) handle specialized tasks independently, (iii) cooperate to satisfy system-level goals, and (iv) achieve a high degree of flexibility. One sub-family DISs is *sensor network systems* (SNSs), which aim at collaborative signal (information) processing on the basis of large-scale, distributed macro- and micro-sensor technologies and connectivity (transmission and networking) technologies. Other sub-family is *intelligent agents systems* (IASs), which manifests in dynamically changing, functionally decentralized, networked multi-agents enabled environments of high robustness and scalability such as distributed energy systems. In some publications, systems with these characteristics are also referred to as distributed autonomous decision making systems.

All of the above mentioned disciplines and system concepts are pointing towards a higher level of integration between the material world and the cyber world. Striving after the highest possible level of it gives the objective for the paradigm of *cyber-physical systems* (CPSs), which is sweeping the society since 2005. As discussed later, CPSs feature extensive functional integration, increasing complexity, emergent intelligence, adaptive structure and behavior, and make a huge impact on humans and the environments. In CPSs, human users can be both in- and out-of-the-loop. The phrase 'cyber-physical systems' has been introduced in the USA by the NSF. As a counterpart of this, systems with practically congruent characteristics have been called *collaborative adaptive systems* (CASs) in Europe. CASs differ from the current generation of open control systems in two important aspects, namely in terms of collectiveness and multi-scaling. They typically comprise very large number of multi-objective units, which have autonomy in their own individual properties, objectives and actions. Decision-making is highly dispersed and the variety of interactions amongst the units may lead to the emergence of new and/or unexpected phenomena and behaviors. The concept of CPSs should be demarcated from that of the Internet of Things (IoT). IoT assumes that things interact and exchange information, and that gives a basis for future pervasive computing environments. However, its objectives are more infrastructural, than application orientated.

The paradigm of CPSs is still in evolution. Therefore, we may come across with rather different interpretations and forms of implementations. According to the classical NSF definition, CPSs are 'physical and engineered systems whose operations are monitored, coordinated, controlled and tightly integrated by a computing and communication core at all scales and levels'. The cyber sub-system is responsible for computation,

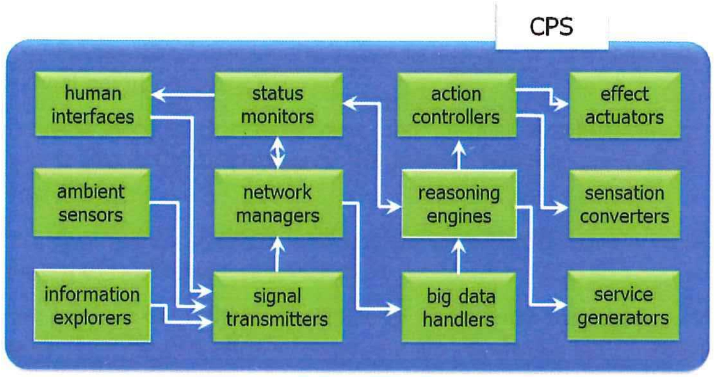

Figure 3 Generic architecture of cyber-physical systems

communication and control, and is discrete, logic-based and event-oriented, while the physical sub-system incorporates natural and human-made components that are governed by the laws of physics, and that operate in continuous time. We should mention that this definition is already deemed as a somewhat conservative one nowadays.

A previous paper proposed a definition that gives more consideration to decentralization, dynamism and evolutionary nature of CPSs. This definition circumscribes CPSs as structurally and functionally open, context-sensitive, intelligent and self-adaptive engineered systems in which the physical and the cyber words synergistically interact and evolve cooperatively, and which gradually penetrate into the social world, as well as into the mental world of humans. *Structural openness* means that they may include collaborative sub-systems of varying spatial and complexity scales in time and space. *Functional openness* implies that they may consist of units that happen to enter or leave the collective at any time. The units (i) can be highly heterogeneous (computers, agents, devices, humans, networks, etc.), (ii) may operate at different temporal and spatial scales, and (iii) may have different (potentially conflicting) objectives and goals. A generic architecture of cyber-physical systems is shown in Figure 3. Many authors differentiate low-end implementations and *high-end implementations*. The former ones are sensing, ubiquitous and smart computing enabled, distributed, networked, collaborative, and proactive (often embedded) systems. The latter are largely complex, open, multi-scale, heterogeneous, intelligent, self-managing, and partly autonomous (reproductive) systems. According to our interpretation, there is a functional synergy among the physical and the cyber components in CPSs. The interactions among remote components happen in real time, under emergent constraints, and often towards non-predefined objectives. Combined with structural variability, these characteristics introduce uncertainty that is difficult to handle by traditional design methods and implementation technologies. The main source of uncertainty originates in the capability of CPSs to change their structure and behavior by learning and adaptation in operation.

Main features of cyber-physical systems

As indicated above, cyber-physical systems have a large number of specific characteristics. Some of them are also characteristics of other systems, but the whole set of these characteristics can be recognized on those systems only that belong to this distinct family of CPSs. Below is this distinguishing set of characteristics listed:

C1 CPSs are designed and implemented in order to support human activities and well-being by *decentralized cooperative problem solving*, in harmony with the techno-econo-social environment,

C2 CPSs consist of a digital cyber-part and an analog physical-part, which are supposed to work together towards the highest possible level of *functional and structural synergy,*

C3 CPSs are *functionally decentralized and geographically distributed open* systems with blurred overall system boundaries,

C4 CPSs are capable not only to dynamically reconfigure their internal structure and *reorganize their functionality/behavior*, but also to change their boundaries,

C5 CPSs are constructed of very *heterogeneous sets of active components*, which can enter and leave the collective at any time, and may encounter other systems with similar or conflicting objectives,

C6 CPSs, as well as their components, may work in *extreme temporal ranges* (from instantaneous to quasi-infinite, and beyond), and manifest on *various spatial scales* (from intercontinental to nano-scales),

C7 components are *typically hybrid structures*, encapsulating various compositions of hardware (e.g. transformer and actuator) entities and embedded cyber (e.g. software and knowledge) entities,

C8 components may have *predefined, emergent or ad-hoc functional connections*, or all, with other interoperable components at multiple levels,

C9 components may operate according to *different problem solving strategies* (plans) towards achieving the overall objective of the system,

C10 components are *knowledge-intensive* and able to handle built-in formal knowledge, knowledge obtained by sensors, and knowledge generated by reasoning and learning mechanisms,

C11 components are able to *make situated decisions* and strive for automated problem solving by gathering descriptive information and applying context-dependent causal and procedural reasoning,

C12 components are able to *memorize and learn from history and situations* in an unsupervised manner and to specialize themselves based on smart software agents and emergent intelligence,

C13 components are able to *adapt to unpredictable system states* or emergent environmental circumstances, as well as to execute non-planned functional interactions and to act proactively,

C14 overall decision-making is distributed over a large number of components (agents), and is based on the *reflexive interactions* among the components and multi-criteria analysis (optimization),

C15 contrary to their distributed and decentralized nature, CPSs need to operate and communicate in *real-time* and in a synchronized manner,

C16 system resources are managed different sophisticated strategies and maintain security, integrity and reliability of the components and the CPSs as a whole.

There have been many possible application domains circumscribed for CPSs such as (i) situated intervention (e.g., collision avoidance), (ii) operation in dangerous environments

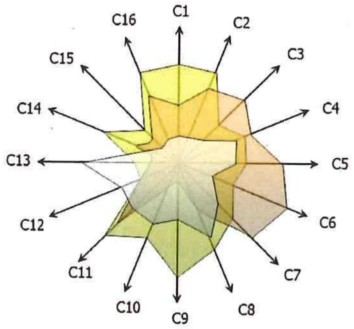

Figure 4 Characteristics profiles of various CPS instances

(e.g., firefighting), (iii) exploration in inaccessible environments (e.g. deep-sea), (iv) precision operation (e.g., robotic surgery and nano-manufacturing), (v) flow coordination (e.g., traffic control, goods manipulation), (vi) efficiency enhancement (e.g., zero-net energy buildings), (vii) augmentation of capabilities (e.g., healthcare monitoring), etc. Actually, only human imagination can be a limit of exploring high-potential applications and innovative solutions.

It has to be mentioned that some implementations of CPSs may not show the entire set of the above characteristics, or may just incompletely realize them. In these cases, we speak about partial compliance with the paradigm of cyber-physical systems. For instance, though it stands in general, CPSs should not always be structurally open, fully autonomic, multi-scaled, or functionally decentralized. The distribution and the measure of partial compliance can be graphically represented and analyzed based on a so-called *characteristics profile diagram*. A case-independent one is shown in Figure 4. By defining qualitative requirements or quantitative criteria for each of the system characteristics, it can be decided if a particular system concept or implementations can (or should) be considered a CPS, or to what extent it complies with the paradigm of CPSs. It has to be mentioned that, in addition to the above characteristics, many researchers have already argued that,

C17 next-generation (molecular and bio-computing-based) CPSs can be supposed to have some level of reproductive intelligence.

Towards social-cyber-physical systems

We are approaching a point where CPSs cease to be just technical systems. They are progressively becoming part of the socio-technical fabric of society. They strongly interact with the human domain and the embedding environment, even if it not always happens in an explicit form. These form two interrelated dimensions of socialization. Therefore, they should be seen as complex socio-technical systems, in which human and technical aspects are massively intertwined. *Social-cyber-physical systems* (SCPS) should work, on the one hand, according to the expectations of humans, communities and society, and on the other hand, under the constraints and conditions imposed by the embedding environment. However, no matter how good the original design specification was, systems become less well adapted to users and environment over time due to changing requirements of the changing users or environment, or to the evolution of the system itself. Therefore, SCPSs are supposed to flexibly adapt to the environment, and to the (communities of) users. These can be achieved based on situation cognizance and context awareness (Figure 5).

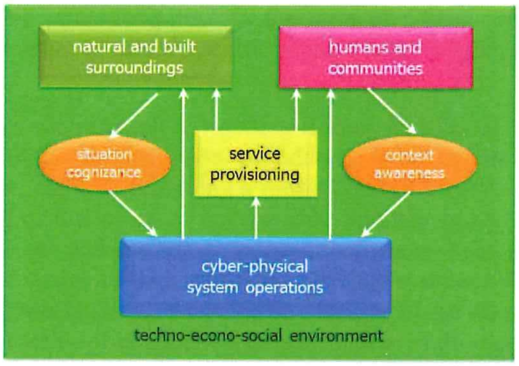

Figure 5 CPS operations in social contexts

Nevertheless, current technological limitations make CPSs intrusive. They are more syntactic, than semantic - therefore they create a mismatch with regard to the human way of thinking and doing. As Biamino discussed, SCPSs should have some basic social abilities such as: (i) detecting users and the social connections between them, (ii) accessing users' data, (iii) inferring the *social context* according to users' networks topology, preferences and features, (iv) inferring social goals according to the social context and the user model, (v) coordinating their behavior, and (vi) providing a context-driven output. The awareness of SCPSs should extend to the intangibles of social context, which includes social culture and norms, personal believes and attitudes, and informal institutions of social interactions. In this context, four additional system characteristics can be stated:

C18 Overall, SCPSs are able to become *aware of the users* and their personal and social contexts, and to adapt themselves towards and optimal symbiosis.

C19 SCPSs are able to achieve the *highest possible level of dependability* (trustworthiness and confidence), accountability, security, accessibility, and maintainability.

C20 SCPSs strive for operating as a *self-organizing holarchic open systems*, with a minimal environmental impact and sustainability from ecological, economic and social viewpoints.

C21 SCPSs are able to achieve a *balance between overheads and outputs*, demand and usage of resources, and wastes and gains.

Major challenges at designing cyber-physical systems

In the context of mechatronic systems, a synergistic integration of mechanical engineering, power electronics and intelligent computer control proved to be the largest challenge, together with the ever increasing desire to improve the performance to cost ratio by engineering design. The design (modeling and simulation) and development (prototyping and testing) of advanced mechatronic systems involve the following areas of specialty: (i) mechanical hardware components (ii) computing and logic hardware components, (iii) computing and control software, agents and object components, and (iv) signal, data, information and knowledge components. The limitations or lack of multi-disciplinary design methodologies have been known to be large challenges for effective design of advanced mechatronic systems. This problem was further articulated by the issues concerning (i) software specification and development, (ii) optimization of control strategies, (iii) elicitation and handling of signals, data and information, and (iv) physical and semantic interoperability. In the case of CPSs, completely new design challenges appeared due to their specific operational, implementation, and usage characteristics.

As generic objectives for research: (i) getting deeper insights in the required synergy between the cyber and the physical parts, (ii) addressing the dynamism and evolutionary nature of the systems, and (iii) providing a unified design theory and methodology that facilitates addressing of the issues of both worlds simultaneously have been identified. This paper intends to address *nine major challenges* that are associated with conceptualization and design of high-end cyber-physical systems. Though closely related to the design issues, two challenges cannot be discussed here due to space limitations. These are: (i) the necessity of conducting transformative multi-disciplinary research for design knowledge synthesis, and (ii) the possibility of using framing methodologies for technological affordances enabled radical innovation of CPSs in social contexts.

Handling aggregative complexity

In general, the term 'complexity' is used to characterize something with many parts in an intricate arrangement. In the context of CPSs, we can differentiate five types of complexities: (i) static complexity (the number and relationships of components that do not change with time), (ii) dynamic complexity (the number and relationships of components that change with time), (iii) self-organizing complexity (open systems reorganize themselves to different systems), (iv) evolving complexity (open systems evolve through time into different systems) and (v) co-evolving complexity (two-way interplay between the changing system and its environment). When all these types and forms of complexities are present, we talk about *aggregative complexity*. Objectives of complexity science are: (i) getting cross-disciplinary insights into complex systems, (ii) explaining emergent structures and self-organization, (iii) generating effective abstractions and models, and (iv) providing control methods for complex systems. Current knowledge offered by complexity science is in its infancy and unable to explain how to reduce and manage aggregative complexities.

It has to be mentioned that static complexity significantly increased by the integration of multiple sub-systems of *various physical scales*. Figure 6 shows the scales which can eventually be integrated into hybrid-scale CPSs. Composition of multiple-scale systems is difficult due to the interfacing problem caused by different physical sizes of the sub-systems and the matching problem that is caused by the different information contents to be processed. Thus, complexity of CPSs increases with: (i) the number of (real or potential) functional components, (ii) the complexities of the distinct functional

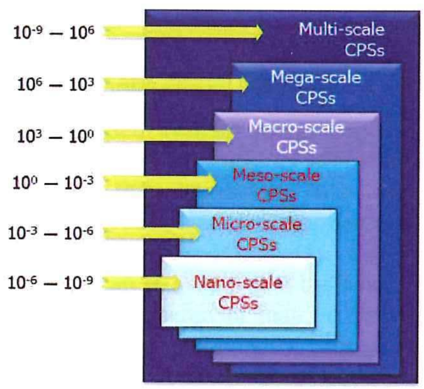

Figure 6 Physical scales of sub-systems that can be included in CPSs

components, (iii) the heterogeneity of the structural components, (iv) the multiplicity of scales of the structural components, (v) the number of connections between them, and (vi) the complexities of the connections among components. Structural complexity is typically accompanied by functional complexities for which temporal predictability is essential.

Traditional divide-and-conquer strategies have not proved to be useful to resolve structural and functional complexities. Complexity theory states that critically interacting components self-organize to form potentially evolving structures exhibiting a hierarchy of *emergent system properties*. It is a challenge in itself because of the difficulty of: (i) capturing and modeling all components and interactions, (ii) managing interactions when everything affects everything else, (iii) considering all potential (non-linear) system states, (iv) quantifying risk/uncertainty for very integrated systems, (v) handling disturbances safely and effectively, and (vi) working with rigid design constraints and tight design space. It is argued that complexity can be handled in design through: (i) structural, behavioral and/or modeling abstractions, and (ii) functional and structural simplifications (considering patterns).

High complexity is often reflected on the non-linear behavior of CPSs. While in case of a linear system the effects (outputs) are proportional to their causes (inputs) and subject to superposition, the behavior of a *non-linear system* cannot be expressed as a sum of the behaviors of its parts (or of their multiples). Open CPSs maybe strongly non-linear systems (not obeying the superposition principle) due to their emergent behavior. Non-linear complex systems cannot be understood by simply decomposing them into components which are added or multiplied together, and are also hard to model and prototype. Non-linearity of CPSs is the result of one or more causal loops (or system learning and adaptation). In every loop the effects or outputs are fed back into the causes or inputs of the process.

Static and dynamic compositional synergy

Composition is becoming a generic design and implementation principle in engineering disciplines. Nowadays, CPSs are conceptualized, designed and implemented by exploiting the benefits of component-based approaches. *Component-based design* (CBD) involves the creation, integration and re-use of hardware, software and knowledgeware components. Components should have rigorous interface specifications that are rich enough to cover all phases of the life cycle. A system design is obtained by assembling strongly encapsulated components according to these specifications. CBD does not prescribe a particular architecture; rather it is defined by the set of components that are chosen and the manner in which they are composed. Reusability depends on the interfaces of components. For hardware parts, adaptable physical modules (modularization) represent the reusable components. For software parts, software frameworks, objects, architectures and design patterns have been developed.

The feasibility of component-based system design depends on two key conditions: composability and compositionality. *Composability* expresses that component properties are not changing as a result of their interactions with other components within the system. It is a measure of the degree to which components can be assembled in various combinations to satisfy specific user requirements. *Compositionality* determines if synergic system-level properties can be established by local properties of components. A CPS is compositional if its emergent behavior may be derived from the behavior of its constituent components. Lack of compositionality causes systems that do not behave well outside a small operational envelope. It is known that CBD helps manage complexity, increases dependability, decreases time-to-market, and optimizes costs, but the principles and methodologies for compositionality in heterogeneous systems are not explored yet.

A large challenge for CBD stems from complexities, in particular from self-organizing and evolving complexities. The overall characteristics of CPSs is the sum of those characteristics that are determined by the designers in the stage of conceptualization and detailing, and those that emerge during operation or are acquired by the systems themselves through self-awareness, self-learning, self-adaptation, self-repairing, and self-sustaining. Designing the results of the *self-** operations brings in a large uncertainty. The principles of how to forecast the emergent characteristics and behavior, and how to integrate, regulate and benefit from them are hardly known now. At the bottom line, the question is how can we architect and engineer CPSs with evolutionary capabilities and under varying operational circumstances to ensure purposeful and secure behavior? In this context, we have to differentiate hard-architected and soft-architected systems. In case of hard-architected systems the execution of the functions happens according to a predefined workflow, while in case of vague-architected systems composition and execution of the functions happens at runtime together with the changes in the system architecture. Dynamic operation of a soft-architected system involves dynamic composition for changing objectives, contexts or situations. Hence, designing of soft-architected systems for autonomy always goes with huge uncertainty and requires the designers to reason with partial, incomplete, inaccurate and noisy information. Dynamic composition also assumes adaptive standardized interfaces.

Dynamic and evolutionary operation in time

The operation of CPSs is event and information driven, but happens in time. In the time domain, operation of CPSs can be interpreted in two dimensions. One is *dynamic operation in time* (DOT) and the other is *evolutionary operation in time* (EOT). Both the physical components and the cyber components perform DOT, but differently. Physical components (e.g. an electromechanical actuator) operate in a time-continuous manner, though showing discontinuities in the characteristics of the physical behavior (e.g. a singularity at reaching to and reverting at a limit state). Cyber components (e.g. a processor) operate in time-discrete (event-triggered) manner. Therefore, abstraction and *explicit time management* (ETM) are needed to bring these operation modes into a common timing framework. As a fundamental concept in CPSs, ETM raises the need for (i) time-oriented modeling of the inherent physical properties, (ii) time-oriented programming and management of concurrency, and (iii) real-time parallel computing. These are, however, not supported well by current programming and computing means.

In decentralized real-time CPSs systems, the correctness of operation depends not only on the logical results of computation, but also on the time at which the results are available. For a time-aware (real-time) procedural integration of the cyber (discrete) world with the physical (continuous) world, both *task-oriented scheduling* (TOS) and *event-oriented programming* (EOP) have to be considered. A task is a logical unit of operation to achieve a particular objective. TOS controls the timing of microprocessors in order to execute a task within a given timeframe and before a deadline. A multiprocessor scheduler decides not only on which task to execute next, but also on by which processor to execute it. Current algorithm-oriented or object-oriented programming approaches do not consider time as a crucial factor, whereas time is perhaps the most important aspect in CPSs.

It has been understood that there is a controversy between various interpretations, e.g. event-driven vs. time-driven operation and control flow vs. data flow, at the specification of operational. This reflects the previously mentioned gap between continuous and discontinuous systems modeling (e.g. state machines and data flow diagrams). The overall operation of CPSs can be modeled as a discrete-event system, which consists of an event record with the associated time-stamp. EOP is based on the assumption that control can be extended to emergent events and unscheduled event interactions in the operation flow of the system. Therefore, usually a two-phase event-oriented control is implemented, in which the first phase focuses on event detection and the second phase deals with event handling, assuming quasi-real-time-computing. Angelov at al. argued that discontinuous event-driven systems can be specified and implemented as time-driven synchronous state machines. Event-driven architecture can complement *service-oriented architecture* (SOA) because services can be activated by triggers fired on incoming events.

One of the largest challenges seems to be designing for self-learning and self-adapting over time and for *semi- or fully-autonomous operation*. This is an important issue since the functionality of the systems with this capability will only be partially determined in the design phase. Designers have to consider not only those functionalities that are specified by the design requirements, but also those that emerge during a longer term operation and use of the designed system. How to consider and manage the effects of long-term learning and adaptation of the system, which depend on (i) the system's evolutionary (self-modifying) capabilities, (ii) the decentralization of determining the objectives, (iii) the interactions with the embedding environment, and (iv) on the conduct of system-level operation and behavior? This requires functional forecasting or behavior prognostics design methodologies that are currently unknown. In addition, it requires new notions of functional correctness and reliability with an explicit expression of time.

Multi-abstraction based modeling

The objective of abstraction is to facilitate coping with the structural and functional complexities and heterogeneity of CPSs. We can identify subjects, aspects and levels of abstraction. Subject of abstraction can be: (i) a system of systems, (ii) a particular system, (iii) a sub-system and (iv) a component. *Abstraction* can be applied, among others, from the aspect of (i) architecture (platform), (ii) procedure (operation), (iii) hardware, (iv) software, (v) networking, (vi) interfacing, (vii) programming, and (viii) computation. The levels of abstraction can be: (i) entity, (ii) group, (iii) neighborhood, and (iv) cluster abstraction. From the viewpoint of components, architectural abstractions can be top down (supporting composability), or bottom up (supporting compositionality). Logically, a component is a sub-system characterized by an abstraction that is adequate for composition and re-use. Semantically, a component is the superposition of a structural model, a behavioral model and an interaction model. *Component abstractions* ignore implementation details and describe properties of components relevant to their composition, e.g. transfer functions, user interfaces. As explained by Lee, components at any level of abstraction should be made predictable and reliable, and the next level of abstraction should compensate for the lack of robustness on a lower level of abstraction.

Abstractions require conceptual modeling, and are either made by simplification (reduction) or generalization (aggregation). In itself, it raises the issues related to the efficiency of model-based conceptualization and development of CPSs and the reliability of *large-scale* system models. Modeling complex CPSs from multiple aspects and various levels requires sophisticated approaches. Again, complexity, scales and heterogeneity influence not only the efforts that are needed for the development of comprehensive models, but also the accuracy and dependability of the models. Though several abstraction-refinement based techniques have been developed and certain automatic abstraction methodologies have already been proposed recently, a general theory and methodology of abstraction in CPSs is not available. The failure of multi-layer abstractions has already been addressed by many researchers. In order fully realize the potential of CPS, the core concepts and abstractions of computing need to be rethought. Lin et al propose that *agent-based modeling* shows promise in overcoming these challenges, due to the flexibility of software agents as autonomous and intelligent decision-making components. Semantic agent systems are even more capable, as the structure they provide facilitates the extraction of meaningful content from the data provided to the software agents. This is in line with the fact that modeling of functional and behavioral CPSs are results of multi-domain synthesis and multi-disciplinary knowledge and indicate the needs for semantics mapping research.

System integrity and behavior verification and validation

Investigation of dependability, reliability and maintainability needs new time-sensitive simulation and evaluation methodologies. These methodologies are still either in a premature stage or non-existent at all. There are however, various principles and analogies that could be used as a robust basis for approaches dedicated to high-complexity CPSs. One such possible approach can be *wavelets of critical system features*. This principle is already known and exploited in other fields such as in some areas of image recognition, biological imaging, communication, remote sensing and other fields of science, and gave floor to the evolution of various wavelet techniques. Wavelets are the underpinning concept of a mathematical method for isolating the most relevant pieces of information in an image or in a signal of any kind. In addition to their large number of neutral (common) components, a large-scale complex CPS may be characterized by its critical conceptual, functional, structural, etc. feature components. Neglecting the neutral components, a much more compact, at the same time more expressive description of the system can be generated. T

he wavelet theory can be extended to recognize influential and critical features of complex CPSs. *Coarse wavelets* can actually be used for identifying general features and *fine wavelets* for identifying particular details. This way, system wavelets can lead to a design and engineering situation in which we don't need all the details in order to learn something useful. Wavelets allow recognition of the features in their details, but from the viewpoint of the system as a whole. System wavelets can be the basis of searching for critical components of a system. In other words, wavelet information may help recognize features that are critical from integration, performance, safety, reliability, maintenance control and similar points of views. Wavelets lead to a natural abstraction and can be a very powerful tool for the analysis, simulation and evaluation of just-conceptualized or already existent systems.

A wavelet-oriented thinking is important since the vast majority of the hundreds, thousands or possibly millions of components of large-scale systems do not play any significant role (and is not able to provide any useful information about the operation of the system). Certain components just assist the operation of others (e.g. multiplexers and distributors), are repetitions of others (sensor components in a network), or are tested or controlled components. Wavelet mapping can be done from multiple aspects, and a multi-aspect analysis of a system can be set up by an *overlaid fabric of wavelets*. The obvious benefit is that the system behavior can be represented and communicated in a much more compact and purposeful way. The wavelet-map will result in zero values for those properties (e.g. lack of reliability) that are not present or influenced at certain components. In combination with multi-level abstraction and multi-aspect investigation opportunities, this approach also supports reverse engineering of the systems. This is exactly how animated movie characters are built up out of wavelets.

Dynamic scalability towards meta-systems

Scaling is about the specification of the properties, control and behavior of CPSs as their size is varied. In case of simple linear systems, scaling would mean scaling would mean application of certain scaling laws. Scalability may be *contraction* (down-scaling) or *expansion* (up-scaling). In the context of CPSs we usually face the problem of up-scaling that assumes the system to have the ability to be enlarged or to handle growing amounts of work in a regular manner. The primary design question is how to architect a system to be extendable to multiple arbitrary scales in time and space. In case of complex systems it is challenging due to the exponentially increasing number of functional relationships among a linearly growing number of distinct components. A system, whose performance increases to the requested level proportionally to the capacity added, is considered to be a scalable system. If the system fails to achieve it, it does not scale.

Scalability is a compound for extending systems. Various *forms of scalability* have been identified and studied in the literature. Functional scalability is about enhancing the system by adding new functionality at minimal effort. Geographic scalability involves maintaining performance, usefulness, or usability regardless of expansion from concentration in a local area to a more distributed geographic pattern. Loading scalability means expanding and contracting the resource pool to accommodate heavier or lighter loads or number of inputs. Administrative scalability concerns increasing the number of users or organizations to easily share a single distributed system. Finally, instrumental scalability is enhancing the ease with which a system or component can be modified, added or removed. Awareness of these forms is important not only because of the various possible objectives, but also for the reason that up-scaling (or down-scaling) implies the necessity of considering all of the above aspects.

The real design challenge is the elaboration of high confidence dynamically extendable and configurable systems. The principles and methods of *large-scale dynamic scalability* are not known. In simple terms, dynamic scalability means an in-process redefinition of the system. Most of the large-scale CPSs will actually manifest as hierarchical 'systems of systems'. They include, as examples, smart energy networks based on micro-production, human well-being service systems, region-wide transportation infrastructures, and catastrophes forecasting networks. From both technological and societal perspectives, the main issue is how to identify and give meaning to interactions within these highly complex, semi-autonomous, cooperative and dynamic systems.

Transformation of big data

According to Gartner, '*big data*' are high-volume (increasing amount of data), high-velocity (speed of data in and out), and/or high-variety (range of data types and sources) information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization. The term 'big data' is used for a collection of structured and/or unstructured (raw) data sets which are produced by installed physical detectors (e.g. cameras or microphones), wireless sensor networks, mobile sensing devices, remote (aerial) sensing, chemical and biological sensors, radio-frequency identification readers, software sensors, or software logs, or any other data generation facility of a cyber-physical system. This is very is combined, or even blended with data that are included in various (socially constructed) open repositories or that is freely available on the World Wide Web. Big data assets are such as document text, images, video, sound, measured values and even three-dimensional object models. Big data is seen simultaneously as: (i) overwhelming amounts of data, (ii) divers data from a variety of sources, (iii) data for semantic fusion, and (iv) data to be abstracted.

According to previous studies, the challenges of big data are difficult to categorize, primarily because the aspect of looking at big data varies. However, we can identify two major challenges related to big data: managing and exploitation. The size of big data sets is beyond the processing ability of software tools commonly used to capture, manage, and process data within a tolerable elapsed time. Most cases big data is notable not because of its size, but because of its relationality to other data. Exploitation of big data means an automated discovery and understanding of deeply embedded facts and meanings by specific *mechanisms of big data computing*. Extracting or synthesizing useful information necessitates transforming raw data both quantitatively and qualitatively in different contexts. Various search and reasoning engines of CPSs apply computer learning mechanisms for these tasks.

Designers should be aware of the kind of big data produced by the active components of CPSs, the target information soaked for, as well as the learning techniques that can provide an efficient transformation in particular contexts. They should make decisions on a scalable and distributed data management, i.e. how the data will be collected, stored, filtered, analyzed, structured, extracted, or annotated within a tolerable elapsed time. However, big data items are usually unrelated and it is difficult to structure them for relational databases. Making sense of them does not come straightforwardly out from the process of data generation. Hence, patterns hiding in massive sets of big data are the primary target. Typically, the design challenge is in the planning of discovery and extraction processes that are able to process enormous data sets in real time or in a reasonable amount of time, and to construct the adaptable automated analysis mechanisms. Typical *automated or semi-automated processing mechanisms* are such as (i) data mining, (ii) specialized documents crawler, (iii) statistical analysis, (iv) pattern matching, and (v) data abstraction techniques. Different techniques are needed for a real-time data analysis and a retrospective data analysis of longer time. The extracted or synthesized information either remains internal for the system to support decision making of the agents, or becomes external (output) information for the system users after structuring and visualization. Literature seems to be in agreement on the fact that only the actionable parts or the extracted information should be transmitted between the modules of CPSs. Future handling of big data demands cost effective, innovative forms of information processing for enhanced insight and decision making.

Testable surrogate prototyping

Though model-based testing and virtual engineering are effective approaches of systems engineering, they are not able to support all aspects of verification and validation of CPSs. Due to the lack of dedicated prototyping methodologies and means early prototyping of CPSs is complicated. Many characteristics, e.g. geographical distribution, diversity and number of components, interaction of multi-scale sub-systems, and operation according to numerous possible scenarios cannot be investigated with the traditional engineering simulation and optimization resources. Traditional virtual simulation methodologies pose many limitations, even in case of a reduced scale empirical testing of large-scale CPSs, due to the needed long preparatory times and huge efforts. Furthermore, they cannot handle *heterogeneous multiple-scale systems* where every scale is different in nature and explore secondary behavioral effects (e.g. interference) that may influence the integrity of system performance. An early rapid prototyping methodology is needed that (i) complements the conventional technologies, (ii) enables the investigation of dependability, functional integrity, technical feasibility, accuracy, etc., (iii) reduces development time and costs, and (iv) allows testing many other factors, as a function of design variables. This methodology is also supposed to go beyond virtual engineering of CPSs by applying a *correct-by-construction* system design methodology and resources.

Surrogate prototyping of complex system (SPCS) seems to be a promising concept for verification and validation of complex CPSs. SPCS is a hybrid testing approach, which extends to the hardware, software, knowledge and reasoning components. It is in line with the principle of CBD, as well as with that of multi-level abstraction. It simultaneously addresses the physical architectures, the computational models, and the process execution in time. SPCS creates approximation models, known as surrogate models, which (i) can represent all kinds of components with sufficient fidelity, (ii) allow the investigation of the interactions among them, and (iii) are cheap(er) to realize physically and to evaluate computationally. The objective of SPCS is to realize high-fidelity workable implementations that are testable from both functioning and utility aspects. In this regard it improves significantly on both purely numerical and purely empirical approaches. There are functional or structural multiplicities and patterns of repetitive components in most of CPSs that can be subject of simplification up to the level of high fidelity. In order to be efficient, SPCS capitalizes on these *simplification possibilities* offered by functional and structural similarities, extent of behavioral influence, and abstraction opportunities of sub-systems and components. Surrogate prototypes can be used equally well for design optimization as well as for design space approximation (emulation). Following the principle of CBD, surrogate models are constructed in a bottom-up fashion. SPCS allows both partial (mock-up type) and all-embracing prototyping. It also applies strategies to discover sources of malfunctioning. Currently SPCS is still in a premature stage. We do not have any theory to explain how high fidelity surrogate prototypes of *large-scale non-patterned CPSs* should be constructed and what criteria/measures to apply to judge their sufficiency.

Rigorous social compliance investigations

As socially relevant, current literature identifies design issues such as (i) balance between privacy and availability, (ii) cyber and physical security, (iii) access control and intrusion detection, (iv) encryption, key management and secure protocols, (v) intelligent informing and classification, (vi) data mining for the detection of physical and cyber-attacks, and (vii) verification and validation in social contexts, but it is not so explicit on the self-capabilities of the systems towards social and environment adaptation. It is argued by Poovendran that CPSs change the notion of physical systems (e.g., vehicles, machine tools, consumer durables) by including humans and the environment in a system-of-systems framework, and thus creating a uniquely large scope and context in which the system behavior must be predictable and provable. The focus of current research gradually shifting from the integration of a large number of homogeneous and heterogeneous systems and creating interfaces among them to providing a knowledge model for *social context-awareness and reasoning* by using ontology-based context modeling, various typified user models, and exploiting of social networks. The system-of-systems thinking urges the development of social context ontologies as the center of the context modeling.

Cyber-physical systems are also penetrating into human cognitive processes. Hence they should also be studied from the perspective of living with and cooperating with CPSs. For instance, recognizing patterns by humans and generalizing them into models are not well understood and not implemented in computers. This also recalls the interface development issues, as well as the need for new insights in the motor, perceptive, *cognitive and affective cooperation* of humans with these systems. Typical form of interaction with a branch of CPSs is tele-operation that manifests in a remote and distributed communication and manipulation. This is applied, for instance, in the case of networks of robots and sensors that work in a cyber-physical space with a remote human in the loop to accomplish dangerous, unpleasant, or super-human activities. Minimal intrusion to human and environment can be facilitated by enrichment of system operation by agent-based smartness. There are *huge knowledge gaps* in these contexts, as well as challenges such as overseeing complexity, real time information provisioning, etc. The exponential proliferation of cyber-physical systems, which is afforded by Moore's law, is not matched by a corresponding increase in human ability to consume information! Therefore, future requires designing for social and mental symbiosis.

Conclusions

Proposition 1: In general, we are still a considerable way away from having a *transdisciplinary theoretical framework* for true CPSs and SCPSs, or even from elucidating the major principles by which they should operate.

Proposition 2: Several definitions of CPSs have been published and many systems have been realized, but design, implementation and utilization of these systems are *still perplexing*, not to mention their possible impacts on the society and the future implications.

Proposition 3: There is no *consolidated design methodology* known that could provide answer to the discussed design challenges, systematize the consideration and management of the effects of long-term learning and self-adaptation of CPSs, and to explain the principles of designing for semi-autonomous or fully-autonomous operation.

Proposition 4: There is a huge knowledge gap concerning the *design and engineering principles and technologies* of realizing high-end, non-linear CPSs (or sub-systems and components) that are compositional, scalable, interoperable, and evolvable.

Proposition 5: New *abstraction methodologies*, as well as pre-implementation modeling, demonstration, prototyping, and empirical testing methodologies are needed in particular for the investigation of contextualized interactions with the human/social environment.

Proposition 6: Next generation CPSs are envisioned to be a horizontally and vertically heterogeneous system of systems, having some level of *reproductive intelligence*. In order to advance the state-of-the-art, both transdisciplinary insights and multi-disciplinary operative knowledge synthesis are needed.

Acknowledgments

This exploratory research work was conducted as part of the efforts of the Section of Computer Aided Design Engineering of the Faculty of Industrial Design Engineering in the Technology Transformation research program.

References

1. Harashima, F. and Tomizuka, M.: Mechatronics - "What it is, why and how?", *IEEE/ASME Transactions on Mechatronics*, 1996, 1(), 1-2.
2. Kyura, N. and Oho, H.: Mechatronics - An industrial perspective, *IEEE/ASME Transactions on Mechatronics*, 1996, 1(), 10-15.
3. Isermann, R.: Mechatronic systems - Innovative products with embedded control, *Control Engineering Practice*, 2008, 16(), 14-29.
4. Bishop, R. H. and Ramasubramanian, M. K.: What is Mechatronics?, in *Overview of Mechatronics*, CRC Press LLC, 2002, 1-11.
5. Lee, I. and Sokolsky, O.: Medical cyber physical systems, in *Proceedings of DAC '10*, ACM,

generation systems, *Computer*, 1988, 21(10), 10-19.

25. Graham, S. Baliga, G. and Kumar, P.: Abstractions, architecture, mechanisms, and a middleware for networked control, *IEEE Transaction on Automation and Control*, 2009, 54(7), 1490-1503.

26. Baillieul, J. and Antsaklis, P.: Control and communication challenges in networked real-time systems, *Proceedings of IEEE*, 2007, 95(1), 9-28.

27. Montestruque L. A. and Antsaklis, P. J.: On the model-based control of networked systems, *Automatica*, 2003, 39(10), 1837-1843.

28. Lin, J., Sedigh, S. and Miller, A.: Modeling cyber-physical systems with semantic agents, in *Proceedings of the 34th Annual Computer Software and Applications Conference Workshops*, IEEE, 2010, 13-18.

29. Biamino, G.: A semantic model for socially aware objects, *Advances in Internet of Things*, 2012, 2(7), 47-55.

30. Marwedel, P., Embedded and cyber-physical systems in a nutshell, *Knowledge Center Article*, DAC.COM, 2001, pp. 1-21.

31. Martín H. J. A., de Lope, J. and Maravall, D., Adaptation, anticipation and rationality in natural and artificial systems: Computational paradigms mimicking nature, *Natural Computing*, 2009, 8(4), 757-775.

32. Gossler G. and Sifakis, J.: Composition for component-based modeling, *Scientific Computing and Programming*, 2005, 55(1-3), 161-183.

33. Benveniste, A. and Berry, G.: The synchronous approach to reactive and real-time systems, *Proceedings of the IEEE*, 79(9), 1991, 1270-1282.

34. Bryant, R. E., Katz, R. H. and Lazowska, E. D., Big-data computing: Creating revolutionary breakthroughs in commerce, science, and society, *White Paper*, Computing Community Consortium, 2008, 1-7.

35. Kay, J. J., Ecosystems as self-organizing holarchic open systems: Narratives and the second law of thermodynamics, in *Handbook of Ecosystem Theories and Management*, CRC Press - Lewis Publishers, 2000, 135-160.

36. Kranzberg, M., Technology and history: Kranzberg's laws, *Technology and Culture*, 1986, 27(3), 544-560.

37. Boyd, D. and Crawford, K., Six provocations for big data, in *Proceedings of the Symposium on the Dynamics of the Internet and Society*, Oxford Internet Institute, 21 September, 2011, 1-17.

38. Laurila, J. K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J. and Miettinen, M., The mobile data challenge: Big data for mobile computing research,

39. Ma, J. Wen, J. Huang R. and Huang, B.: Cyber-individual meets brain informatics, *IEEE Intelligent Systems*, 2011, 26(5), 30-37.

40. Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T. and Crombecq, K., A surrogate modeling and adaptive sampling toolbox for computer based design, *Journal of Machine Learning Research*, 2010, 11(7), 2051-2055.

41. Ning, H. and Liu, H.: Cyber-physical-social based security architecture for future internet of things, *Advances in Internet of Things*, 2012, 2(), 1-7.

42. Poovendran, R.: Cyber-physical systems: Close encounters between two parallel worlds, *Proceedings of the IEEE*, 2010, 98(8), 1363-1366.