SICS Technical Report T2012:01 ISSN 1100-3154

SWEDISH INSTITUTE OF COMPUTER SCIENCE

Federated Embedded Systems – a review of the literature in related fields

Avenir Kobetski and Jakob Axelsson

{avenir, jax}@sics.se

Software and Systems Engineering Laboratory
Swedish Institute of Computer Science

2012-01-20

Abstract:
This report is concerned with the vision of smart interconnected objects, a vision that has attracted much attention lately. In this paper, embedded, interconnected, open, and heterogeneous control systems are in focus, formally referred to as Federated Embedded Systems. To place FES into a context, a review of some related research directions is presented. This review includes such concepts as systems of systems, cyber-physical systems, ubiquitous computing, internet of things, and multi-agent systems. Interestingly, the reviewed fields seem to overlap with each other in an increasing number of ways.

Keywords:
Federated embedded systems, systems of systems, cyber-physical systems, ubiquitous computing, internet of things, multi-agent systems.

Table of Contents

1 Introduction

The role of computing devices, embedded into everyday objects, has grown tremendously over the last two decades. To give an example, a typical car produced at the beginning of the 1990-ies was largely a mechanical unit. Today, a large part of the development costs in a typical front-edge car manufacturing company are related to software development.

The unprecedented complexity of existing software systems is paralleled by an analogous development within the hardware technology. Hardware is being developed faster, while it is cheaper and more powerful than ever before. Of course, at the same time hardware devices are becoming ever more complex and heterogeneous.

The rapid growth and success of communication technology is the third constituent of what many consider as the next technological leap facing the human society, namely the emergence of interconnected intelligent things or embedded devices, capable of communicating both with each other and humans, sensing, taking decisions and acting on these decisions. In fact, numerous applications of such interconnected things are already starting to reach the market, for example home care surveillance devices, disaster warning systems, smart energy grids, intelligent buildings, autonomic vehicle convoys, traffic prediction systems, smart automation, etc.

However, the rapid growth in software, hardware and communication technologies is not only an enabler but also a grand challenge for the future interconnected systems. The foreseeable complexity of such systems, together with their inevitable criticality for the human well-being in many applications, pose a large number of challenging questions that cross-cut several research disciplines. In our work, we aim to address some of these questions, with the focus towards federations of embedded systems.

A federated embedded system (FES) is defined as a constellation of devices that are part of and control different products, and that exchange data with each other and with external servers to the benefit of all, in such a way that no individual device is in control over the others. Note that in many cases this implicitly means that the constituent devices are produced by different manufacturers using different platforms, standards, etc. Further, FES need not have a static structure, but are established, reestablished and extended over time, and a particular device can be part of several FES at different times or simultaneously.

Naturally, it must be profitable and secure for an embedded device to participate in a federation. In other words, the efficiency and/or the possibilities of a device should be enhanced, while certain quality attributes of FES, such as performance, safety, privacy and robustness should be guaranteed. To meet these concerns, development is needed within such fields as programming technology, software and hardware architecture, software development methods and tools, business structures, communication protocols, data management and human-machine interaction, to mention a few.

The enormous potential for various aspects of human life that the vision of interconnected smart objects offer, together with its significant technical challenges, has attracted the interest of researchers within different disciplines. The scope of this paper is to highlight some of the most promising research directions related to the notion of FES, not with the aim of providing an exhaustive survey, but rather of attempting to define the fields and to chart associated historical and future directions and challenges. It should be stressed though that for the time being, as the vision of

interconnected smart things is still relatively new, so are the concerned research fields. In consequence, many of the associated research terms still lack globally accepted definitions and are used in a rather ad-hoc manner, with somewhat different meanings in different contexts.

The related research is outlined in Section 2, with one subsection per discipline, while general conclusions are gathered in Section 3.

2 Related research

An initial analysis of the work that in some way relates to the notion of intelligent, interacting, distributed systems revealed a large number of research fields or buzzwords that could be worth further examination. A more careful investigation left us with five important, vibrant, and in some sense related research directions, namely systems of systems (SoS), cyber-physical systems (CPS), pervasive or ubiquitous computing (Ubicomp), internet of things (IoT), and multi-agent systems (MAS). In the following, these research fields will be shortly reviewed.

2.1 Systems of systems

As the complexity of engineered systems has been growing, the notion of *systems of systems (SoS)* has attracted more and more popularity. Partly, this popularity is probably stemming from the apparent transparency of the term itself. Most people have an idea of what a system may look like. Thus, it might seem that extending this understanding to a system of systems should be rather straightforward.

Unsurprisingly, this is a deceptive feeling, which a more thorough study reveals. Although most SoS-definitions found in the literature share some common perception of the meaning of SoS, the actual suggestions for the distinguishing SoS-features, that differentiate them from ordinary systems, are quite varying. In fact, in a report investigating and classifying the usage of the SoS-term, (Boardman, Pallas, et al. 2006) reviewed more than 40 different SoS-definitions.

Some of the first documented SoS-definitions were either too specific or too vague. In 1996, Manthorpe proposed an SoS to be concerned with the interoperability and synergism of military systems, such as Command, Control, Computers, Communications, and Information (C4I) systems with Intelligence, Surveillance and Reconnaissance (ISR) systems (Manthorpe Jr. 1996). One year later, Kotov defined an SoS to be a large-scale concurrent and distributed system, composed of complex systems (Kotov 1997). Next, Lukasik argued that SoS education should involve the integration of systems into system of systems that ultimately contribute to the evolution of the social infrastructure (Lukasik 1998). So far, none of the proposed definitions actually provided a clear way of separating SoS from simply complex systems.

One of the most influential papers within the field, that actually tried to describe distinguishing SoS attributes in a systematic way, (Maier 1998), declared five principle characteristics of such systems:

- Operational independence of the elements – the SoS is composed of systems which are independent and useful in their own right.
- Managerial independence of the elements – the component systems not only can operate independently, they do operate independently.

- Evolutionary development – functions and purposes are added, removed and modified with experience.
- Emergent behavior – the system performs functions and carries out purposes that do not reside in any component system.
- Geographic distribution – the components can readily exchange only information and not substantial quantities of mass or energy.

Also, Maier classified the systems fulfilling the above characteristics into three subclasses:

- Directed SoS – built and centrally managed to fulfill specific purposes, even though the component systems maintain an ability to operate independently.
- Collaborative SoS – lack central authority but have a common purpose. Such systems must, more or less, voluntarily collaborate to fulfill that agreed upon central purpose.
- Virtual SoS – lack both central authority and a pre-defined purpose. This leaves the supersystem relying on invisible mechanisms to maintain some emergent desirable large-scale behavior.

In (Sage and Cuppan 2001), these ideas were formalized into a definition stating that an SoS is a non-monolithic system where all or a majority of the above characteristics are present. In the same publication, the notion of *Federations of Systems (FoS)* was presented as a related concept to SoS, with the distinctive characteristics being a stronger emphasis on autonomy, heterogeneity and geographic distribution of the components (Sage and Cuppan 2001).

The work of (Maier 1998) and (Sage and Cuppan 2001) can probably be seen as the common denominator for the subsequent SoS-definitions. Most of them include variations of some of the above attributes, while imposing and focusing on additional descriptive terms or requirements, see (Boardman, Pallas, et al. 2006) for a review. For example, (Krygiel 1999) states an SoS to be a set of different systems so connected or related as to produce results unachievable by the individual systems alone (emergent behavior). (Shenhar 2001) speaks about widespread collections of networks of cooperating systems (geographic distribution). (Carlock and Fenton 2001) propose the notion of enterprise SoS engineering as a coupling of traditional systems engineering activities with enterprise activities of strategic planning and investment analysis. (Keating, et al. 2003) describe autonomous, heterogeneous, distributed and embedded systems (independence and geographic distribution). Jamshidi combined the ideas of Kotov and Maier into defining SoS as large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal (Jamshidi, System of systems engineering, Innovations for the 21st Century 2008). In (DeLaurentis 2005), Maier's five characteristics are reused and extended with the notion of networks of heterogeneous trans-domain systems.

While providing a rich set of SoS definitions, suitable for different applications and architectural structures, there were still some gaps to fill in the above mentioned bulk of work. The most obvious was, as already mentioned, the lack of a coherent, widely accepted definition of the SoS-term. As a consequence, it was still not always clear how to distinguish between SoS and ordinary systems. In an effort to meet this shortcoming, (Boardman, Pallas, et al. 2006) took a step back and presented a new set of SoS-characteristics (as opposed to definitions) with the focus on carefully declaring features that distinguish SoS from more tightly-integrated systems of parts (or in other words, systems of subsystems or simply systems). In doing so, they based these distinguishing characteristics

upon a review of the combined mass of SoS-related work. The result was the following five characteristics:

- Autonomy – a (constituent) system exists to fulfill its own purpose. When combined into an SoS, the system will normally conform to certain constraints, but it should retain its autonomy. In contrast, if a "system" has ceded its autonomy and no longer has its own purpose, it should be simply classified as a part or subsystem of a larger system. This is related to the operational and managerial independence, coined by (Maier 1998).
- Belonging – constituent systems choose to belong on a cost/benefit basis and because of their belief in the SoS supra purpose. SoS and constituent systems negotiate about the latter's belonging and the former's acceptance to the overall system. Normally SoS can continue functioning even if some constituent systems choose to leave it. Parts, on their hand, do not choose their belonging but are integral to the system, which in turn cannot function without any of its parts.
- Connectivity – possible connections between parts or subsystems are normally defined at design time, generally with the goal of encapsulating and hiding away a large amount of connections within the subsystems. In contrast, to preserve autonomy and the right to choose belonging, this cannot be done to the constituent systems of an SoS. Instead, the constituent systems should have the right to freely and adaptively determine their interfaces according to their own purposes or their views of what is best for the common SoS aims. This idea of evolutionary development of available interfaces is also called open connectivity.
- Diversity – requirement-driven system modeling often involve hierarchic or modular thinking, which preferably leads to a low number of distinct subsystems. This is normally beneficial for conventional systems, helping to keep their complexity down. However, when faced with the evolutionary and uncertain nature of an SoS, the idea of avoiding diversity is no longer adequate. Instead an SoS should be heterogeneous, containing a large variety of functions/systems, able to respond in different and sometimes complimentary ways to the unpredictable challenges on the overall SoS purpose.
- Emergence – in the case of ordinary systems, both intended and unintended (bad) behavior can to some extent be foreseen, tested and restricted at design time. When it comes to SoS, the required functionality is generally not clear at the design time. Thus, the emergence possibilities should not be restricted. On the contrary, emergent behavior should be promoted, which is achieved by the four earlier mentioned SoS-characteristics. If this is done in a proper way, as claimed in (Boardman and Sauser 2006), SoS will quickly detect and destroy unintended behaviors, in analogy with the immune defense of a human body.

Another attempt to revitalize the field and break free from the somewhat confusing mass of SoS definitions was done in (Northrop, et al. 2006). This work developed the notion of *ultra-large-scale systems (ULS)*, also named *socio-technical ecosystems*, based on the software engineering point of view on SoS. The main underlying assumption was that the future systems will be extremely large and complex, not least in terms of lines of code, amount of data stored, number of interdependencies between software components, and number of hardware elements. In addition, people were given a more central role than in many other disciplines, being considered as the elements of the system rather than just their users. With this in mind, (Northrop, et al. 2006) claimed

that our current understanding of software development and social behaviors is not nearly sufficient. Based on the above considerations, the following characteristics of a ULS system were proposed:

- decentralization – the scale of ULS systems means that they will necessarily be decentralized in a variety of ways—decentralized data, development, evolution, and operational control.
- inherently conflicting, unknowable, and diverse requirements – ULS systems will be developed and used by a wide variety of stakeholders with unavoidably different, conflicting, complex, and changing needs.
- continuous evolution and deployment – there will be an increasing need to integrate new capabilities into a ULS system while it is operating. New and different capabilities will be deployed, and unused capabilities will be dropped; the system will be evolving not in phases, but continuously.
- heterogeneous, inconsistent, and changing elements – a ULS system will not be constructed from uniform parts: there will be some misfits, especially as the system is extended and repaired.
- erosion of the people/system boundary – people will not just be users of a ULS system; they will be elements of the system, affecting its overall emergent behavior.
- normal failures – software and hardware failures will be the norm rather than the exception.
- new paradigms for acquisition and policy – the acquisition of a ULS system will be simultaneous with the operation of the system and require new methods for control. No centralized authority will be able to successfully manage an ULS system due to its scale.

A relation to the SoS field is given by the authors themselves, pointing out that the ULS systems share some concepts with the idea of virtual SoS, presented in (Maier 1998).

2.2 Cyber-physical systems

The concept of *cyber-physical systems (CPS)* is relatively new even though the driving forces behind it are not. Looking at the scientific background of the people involved in the CPS field, it appears to stem mainly from such areas as real-time computing, distributed control systems, wireless sensor networks, and mobile systems. This view is confirmed by the challenges and applications, discussed within the CPS community.

The creation of the new research direction began with a series of workshops supported by the US National Science Foundation (NSF), related to the CPS and their possible applications to aviation, manufacturing, medical devices, and software development, see (CPS Steering Group 2008). This led to the importance of CPS being recognized at a relatively high level of US administration (US President's Council of Advisors on Science and Technology (PCAST) 2007) which served as a catalyst for the current highly vibrant research activity within the field.

The PCAST report noted that the NIT (Networking and Information Technology) R&D-portfolio was imbalanced in favor of low-risk (small-scale and short-term) projects, a situation that is paralleled by several European countries, which was indicated as a risk to the long-term position of the USA as one of the field leaders. The recommendation was to substantially increase funding within four selected areas of NIT field, with the area of "NIT Systems Connected with the Physical World (which are also called embedded, engineered, or cyber-physical systems)" holding the first place on the list of four (the other prioritized research areas were software, digital data and networking).

The workshop series that preceded the establishment of the CPS community helped to create a common understanding of the CPS-term. Generally, cyber-physical systems are defined as networked engineered systems that extensively rely on computation and communication technology, and are deeply embedded in and interacting with physical processes to add new capabilities to physical systems, see e.g. (CPS Steering Group 2008), (CPS Summit Report 2008), (Rajkumar, et al. 2010), (Sha, et al. 2009), (Lee 2009). Although the notion of CPS is closely related to such concepts as networked embedded systems or pervasive computing, the founders of the new research community argued that a new term was needed to emphasize the strong focus that should be placed equally on physical, computational and communicational aspects of CPS. They argued that although computer and control sciences have coexisted since the beginning of 1940-ies, not much has been done to bring these fields closer to each other. To address this, the new research community should avoid neglecting the physical reality, such as measurement noise, communication delays, power consumption, disturbances, and inaccuracies in actuation, etc. when designing computer systems. On the other hand, computers should not be conceived as infallible when modeling physical processes.

It is pretty safe to say that most future technical devices will fall into the category of things that are both cyber and physical in some respect. It is envisioned that CPS will play an important role in such areas as future energy systems, transportation systems, health care, manufacturing, living environments, climate monitoring, agriculture, defense, etc. Thus, the number of possible applications and challenges for the CPS research and development community is vast. A number of challenging research topics have been outlined in the literature as a result of the extensive workshop work within the newborn CPS community:

- Composition – CPS are assumed to be highly heterogeneous, both in terms of their components and imposed design requirements. Current compositional frameworks are generally tailored for specific problem areas and not designed to cope with such heterogeneity (CPS Steering Group 2008). Thus, a new theory of system composition is needed. The theory should account for different time scales, location, memory requirements, cost, energy and security requirements of the components. Further, it should be able to describe both deterministic and probabilistic requirements, as well as both time- and event-based systems.
- Robustness, reliability, safety and security – systems will be exposed to unexpected failures, uncertainties in the environment, and adversary attacks, both on cyber and physical levels. This will become even more challenging as the system scale, complexity, and openness rise. Of course, this must be dealt with, especially when it comes to safety critical services. Similarly, upgrade of running devices is unavoidable in the long-run and should be done in a safe way. Adaptability, recovery modes, redundancy, self-organization, and reconfiguration have been mentioned as possible solutions.
- Privacy – the substantial amount of transmitted data that is expected in networks of CPS, in particular time and location related data raises the question about privacy. This is further complicated by the fact that there are often limits on how much information physical systems can hide.
- Trust – related to both security and privacy. How can we trust a system to behave acceptably, both with respect to our purposes and in terms of not revealing more about us than we are ready to accept? Which parts and signals of CPS can be trusted and to what degree? How to handle untrusted sources?

- Decentralization of sensing, computation and control – CPS are inherently distributed, which makes the idea of centralized control inconceivable. Instead, some sort of decentralized control strategies or incentives for desired behavior are needed. Questions arise as to what information should be collected and when; where, how and when should this information be treated; which parts of the systems should be controlled or simply affected by which other parts, etc.
- Verification, validation, testing and certification – verification of timed systems requires generally an exponential effort. However, some sort of "correct-by-construction" approach and online verification of certain key properties are needed. This is even more important for open systems and systems based on wireless communication.
- Architecture – new network protocols must be designed for large-scale CPS. Also, the architectural structure should be able to capture a variety of physical information, while promoting important CPS properties, such as composability, schedulability, component-based verification, safety, and decentralized control.
- Programming abstractions – new programming abstractions will be needed to explicitly capture and control real-time properties of CPS. This might lead to a rethinking of the traditional split between programming languages and operating systems. Model-based development tools are expected to gain ground.

2.3 Pervasive / ubiquitous / context-aware computing

The most profound technologies are those that disappear. With these legendary words began Mark Weiser's article that is often referred to as the starting point of *ubiquitous or pervasive computing (Ubicomp)*, (Weiser 1991). In fact, Weiser's ideas are still valid even today. The main thought is that computing devices are expected to become seamlessly integrated into the everyday world. For this to happen, two main prerequisites are needed. Firstly, the number and density of embedded devices interacting with humans and each other should be large. In other words, computers should be all around us (ubiquitous) and pervade most aspects of the human life. Secondly, the computing devices must become invisible in the sense that they operate without distracting users more than absolutely necessary.

To achieve this vision of non-distraction, the awareness of computing systems about their surrounding contexts has been considered as central. A rather broad and logical definition of context refers to it as any information that can be used to characterize the situation of an entity, where an entity is a person, a place, or an object that is relevant to the interaction between some application and its user (Abowd, et al. 1999). Naturally, the actual choice of which information to consider as relevant is strongly dependent on the application, which in practice has led to a quite sprawling usage of the term context.

While most early context-aware application focused on the user location as the main object of study, (Schilit, Adams and Want 1994) presented a more embracing description of context that can be partitioned into three main context types:

- Computing context (network connectivity, communication cost, available bandwidth, nearby resources, etc.).
- User context (user identity, profile, location, social situation, etc.).
- Physical context (lighting, noise, traffic condition, temperature, etc.).

In (Chen and Kotz 2000), temporal information is proposed as a fourth type of context. More on context-awareness can be found in e.g. (Baldauf, Dustdar och Rosenberg 2007), where a survey of different design principles and context models for context-aware systems is given, or (Abowd and Mynatt 2000) that outlines the research progress during the 1990-ies, together with some thoughts about the future.

When looking back at the history of the computer science, it becomes evident that the idea of pervasive computing came forward as a logical consequence of previous research within distributed systems and mobile computing, (Satyanarayanan 2001). The distributed systems field contributed with the understanding of such questions as remote communication, fault tolerance and security. The mobile computing on its hand laid ground for such technologies as adaptive applications, energy-efficiency and context-awareness. However, while mobile systems were designed to be both context-aware and reactive, the pervasive computing took another step and is often described as a proactive technology, not only concerned with collecting the information about the surrounding context, but also with acting on and adapting to that context, all the time with the aim of blending into the background. The research challenges mentioned in (Satyanarayanan 2001) that trace their origins to the field of distributed computing are the questions of scalability, heterogeneity of components, privacy, and trust.

In (Saha and Mukherjee 2003), the above challenges are reiterated, while pro-activeness is generalized to smartness and an additional challenge of integration is presented. Interestingly for our scope, the idea of federations of components that require some sort of coordination is mentioned as a prerequisite for the integration process. Smartness, on its hand, is defined to go beyond mere algorithm development and to include a deeper understanding of the physical space.

In (Estrin, et al. 2002), the immense scale and extreme dynamics in demand of the envisioned pervasive systems are mentioned as future research challenges. In many cases, passive energy-saving vigilance will be a sufficient task of a pervasive system, while at other times frantic activity, often concurrently with other devices, will be required. How to cope with these opposing types of activities is an open question. Other presented challenges are variability in structure and tasks of the devices and systems, autonomy considerations, systems complexity, etc. Further, there is a need for the development of such technologies as programming models, closed-loop control, predictability, diagnosability, environmental compatibility, system-wide architecture that supports interrogating, programming and manipulating physical world, capability to self-organize, handling of stochastic communication delays, anonymity preservation, and energy harvesting. Interestingly, a number of the above challenges could be attributed to the field of cyber-physical systems, or even to the internet of things, or multi-agent systems, reviewed in the following sections.

To mention a recent publication within the Ubicomp field, (Bardram and Friday 2009) emphasize the following challenges:

- Resource-constrained devices
- Volatile execution environments (how to discover and connect to other devices?)
- Heterogeneous execution environments
- Fluctuating user environments (many-to-many relationships between users and devices)
- Invisible computing (autonomic computing, pro-active computing, graceful degradation)
- Security and privacy

Again, most of these characteristics could be easily attributed to any of the other research directions that are reviewed in our work. For example, resource-constrained devices form the cornerstone of the CPS field. Both autonomy and fluctuating user environments are paralleled in the SoS world by Maier's idea of virtual SoS, and the autonomy and connectivity characteristics of (Boardman, Pallas, et al. 2006). As will become evident in the following, the pro-activeness, autonomy, heterogeneous and fluctuating user environments are quite common properties of multi-agent systems. Finally, the question of how to discover and connect to other devices is central to the concept of internet of things, while the security and privacy considerations are recurring in all of the above fields.

To conclude, the pervasive computing research seems to be a rather human centered discipline. In fact, the notion of invisibility, so prevalent in this field, presupposes that human perception of pervasive systems is taken into consideration. This might explain why the majority of pervasive applications have traditionally been of a rather modest scale, focusing on limited groups of people that interact with the smart objects in their vicinity (Estrin, et al. 2002). However, this focus is starting to shift, with an increasing number of researchers realizing that the future belongs to large-scale heterogeneous systems (Lukowicz, Choudhury and Gellersen 2011).

2.4 Internet of things

According to Kevin Ashton, the term *Internet of things (IoT)* came into being at a presentation he held at Procter & Gamble in 1999 (Ashton 2009). He argued that human beings were limited in their ability of capturing and processing data about the real world. Instead, computers or things should be empowered to be central actors of the future internet. These ideas lead to the creation of Auto-ID Center at MIT, with the vision of a world where

> *"all electronic devices are networked, … using physical tags that allow remote, contactless interrogation of their contents; thus, enabling all physical objects to act as nodes in a networked physical world."* (Sarma, Brock and Ashton 2000).

The Auto-ID Center soon expanded into Auto-ID Labs, a network of research institutions collaborating on a common mission of enabling every physical object anywhere in the world to be uniquely identified. The idea was to use the existing internet structure, upon which object tracking and information sharing possibilities would be added. The main results of this mission were the development of the Electronic Product Code (EPC) standard and Radio Frequency IDentification (RFID) technology. In the last decade, such focus on things and their identity has been one of the major driving forces within the IoT community.

However, the scope of the IoT research and the meaning of the term itself have grown considerably during this time. Even though a consistent definition of the IoT is still missing, the following vision presented by the European research initiative on IoT (Cluster of European Research Projects on the Internet of Things 2010) is quite representative of today's broader understanding of the term:

> *"The Internet of Things is an integrated part of Future Internet and could be defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network."*

With this definition in mind, it is not surprising that the research within the IoT field is considered to be built up of three underpinning legs, namely the "things"-, "internet"-, and "semantic"-oriented

approaches (Cluster of European Research Projects on the Internet of Things 2010) (Atzori, Iera and Morabito 2010). While the "things" part of the IoT research focuses on unique identifiability, the "internet" part is primarily concerned with enabling connectivity of things by means of generic and energy-efficient communication protocols. The "semantic" research has studied questions of how to represent, store, search and organize vast amounts of data that are expected in the global IoT. This research part also includes the notions of intelligence and self-organization.

A core part of the communication research is related to the wireless sensor networks (WSN) technology. This includes such issues as network architectures, communication protocols, energy-efficiency, programming languages, operating systems, security, etc. Several lightweight operating systems, specially tailored for WSN applications, have been developed, with the most popular probably being the TinyOS (Levis, et al. 2005) and Contiki OS (Dunkels, Grönvall and Voigt 2004). When it comes to the communication protocols, the notion of *Web of Things* was recently introduced (Guinard and Trifa, Towards the Web of Things: Web Mashups for Embedded Devices 2009) to represent the reuse and extension of current web protocols, such as HTTP, to the domain of things. This seems as a relatively simple and yet promising idea that may allow to take a leap in interconnecting heterogeneous things. However, a possible drawback of such a general approach is a higher communication overhead (Guinard, Trifa and Mattern, et al. 2011), which may lead to unsatisfactory energy consumption for some real-time applications. For a more thorough review of the WSN technology, the interested reader is referred to (Karl och Willig 2005) or (Yick, Mukherjee and Ghosal 2008).

In a survey paper on IoT (Atzori, Iera and Morabito 2010), a number of enabling technologies were presented. Besides the already mentioned RFID and WSN technologies, the focus was laid on middleware architectures, preferably based on service oriented approaches. Also, trust, privacy, and security issues were mentioned. A number of potential applications of IoT were grouped according to their domain:

- Transportation and logistics, including assisted driving, mobile ticketing, monitoring environmental parameters, and augmented maps.
- Healthcare, including tracking in hospitals, patient identification, staff authentication, automatic data collection and transfer, and real-time health sensing.
- Smart environments, such as comfortable homes and offices, industrial plants, as well as smart museums and gyms.
- Personal and social domain, including social networking, trend queries, loss and theft recovery.
- Futuristic applications, such as robot taxi swarms, city information models, enhanced game rooms.

In (Katasonov, et al. 2008) the semantics-oriented view of IoT challenges is given, with the main issues being the heterogeneity of components, standards, data formats, protocols, etc., and the immense scale of the future IoT. The vision of a true interoperability between components of different types, rather than just interconnectivity, is presented, including such questions as automatic service discovery and emergent functionality. The proposed solutions include semantic and agent-based technologies.

Similar view of the future IoT challenges is outlined in (Haller, Karnouskos and Schroth 2009), where the questions of identification, addressing (including logic addressing based on certain properties), heterogeneity of components, service discovery mechanisms, context-sensitivity, composition of services (possibly using some sort of semantics), and service intermediaries are discussed. Interestingly, the focus of the above paper is not purely technical. It is argued that the future business processes will be decomposed into distributed process steps due to the competitive advantages of a better information supply and faster decision-making that this approach promises. Consequently, the current lack of real business cases is mentioned as one of the main challenges to the IoT.

An interesting elaboration on the current and the envisioned state of affairs within the European IoT research was presented in a recent European Commission report (Cluster of European Research Projects on the Internet of Things 2010). More than 30 research activities within the scope of IoT financed by the European Commission were shortly introduced, together with a strategic research agenda for the years to come. The objectives of the report was partly to foster the collaboration between different research groups, and partly to define the IoT term, identify its application domains and enabling technology, as well to formulate more detailed short-, medium- and long-term research agendas.

While the above report's definition of the IoT was already mentioned earlier, an excerpt from its vision statement further emphasizes the different parts of the emerging field:

> *"Internet of Things hosts the vision of ubiquitous computing and ambient intelligence enhancing them by requiring a full communication and a complete computing capability among things and integrating the elements of continuous communication, identification and interaction. The internet of Things fuses the digital world and the physical world by bringing different concepts and technical components together: pervasive computing, miniaturization of devices, mobile communication, and new models for business processes."*

Interestingly, besides the reference to ubiquitous computing, this vision comprises the physical, communicational, and computational aspects that are often referred to as the distinguishing features of cyber-physical systems.

Returning to the European Commission report, the portfolio of possible applications showed to be quite vast, a seemingly unavoidable consequence of clustering together a large number of research groups. The result was the following list of IoT application domains: aerospace and aviation, automotive, telecommunication, intelligent buildings, healthcare, independent living, pharmaceutical, retail, logistics and supply chain management, manufacturing and product lifecycle management, oil and gas, safety, security and privacy (e.g. earth quake surveillance, building monitoring, equipment and personnel surveillance), environment monitoring, people and goods transportation, food traceability, agriculture and breeding, media, entertainment and ticketing, insurance, and recycling.

Looking through the list of enabling technologies, a number of familiar concepts can be found, including scalability, resource-efficiency, identification technology, decentralized architecture, business concepts, communication technology, mobility and network discovery, heterogeneity of hardware and software, autonomy, adaptability and self-organization, data processing, standardization, security and privacy, etc.

Finally, a research agenda for each technology area, partitioned into time periods 2010-2015, 2015-2020, and beyond was proposed. The medium-long-term research questions included privacy-aware identification, adaptive, context-based or even cognitive architectures, self-organizing networks, goal-oriented software, smart and tiny sensors, context-aware data processing, energy harvesting, self-adaptive security mechanisms, and standards for cross interoperability with heterogeneous networks, to mention a few.

2.5 Multi-agent systems

The idea of autonomous agents and their composition into multi-agent systems (MAS) have been discussed extensively in the computer science society since the late 1980-ies. Nevertheless, also this field lacks a general widely accepted definition of its naming term.

In (Franklin and Graesser 1997), an overview of existing agent definitions up to that date is given, together with a new unifying definition, aiming at being maximally permissive with respect to the previous work, while drawing a distinction between an agent and an ordinary software program. The exact phrasing of this definition is:

> *"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future."*

Apparently, such definition can be applicable to a very wide range of systems, putting the usefulness of the approach in question. To remedy this problem, the authors propose to use more fine-grained classification schemes to describe e.g. communicative agents, mobile agents, learning agents, etc. Besides its slight vagueness, the above definition makes a tight coupling between an agent and the surrounding environment. Thus, if an agent loses its ability to sense or act due to a change in the environment, it is no longer considered as an agent according to the above definition.

Also, with respect to the systems perspective, the above definition is lacking an important property, namely the ability to communicate. Thus, we choose to adopt another well-cited and clearly stated definition (Wooldridge and Jennings 1995), where an agent is said to be a hardware or software-based computer system with the following properties: autonomy, social ability (communication), reactivity (sensing and acting), and pro-activeness (goal-directed behavior).

Note that the question of whether MAS should have a common overall goal is not explicitly mentioned. In fact, early work on MAS focused on common system goals while as time passed, more open MAS architectures, consisting of self-interested agents have become an equally popular subject of study. In both cases, MAS are generally considered as decentralized, dynamic systems. As pointed out in (Jennings 2000):

> *"Agent-based techniques are the ideal computational model for developing software for open, networked systems (such as the Internet). … Open, networked systems are characterized by: no single controlling organization, diverse range of stakeholder interests, constant change."*

As a consequence, interactions between agents are assumed to evolve dynamically and are generally not specified in detail at the design time, even though the structure for such interactions may (and often should) be designed.

When the notion of agents started to gain momentum in the research community, drawing the best features from the object-oriented (OO) programming and knowledge engineering, it was strongly hyped as the next programming revolution. The main reasons were probably the benefits of intelligence and flexibility that the agents promised, together with a solid underlying foundation of the OO-paradigm. In fact, the main difference between the OO and agent-based software development is that agents generally have the right to decide on their own about executing some requested task or not. Another difference is that agents don't really need a request to start execution. If properly used, these characteristics may offer substantial possibilities to adapt in a beneficial way to changing environment. However, on the downside, this makes any system of agents more unpredictable and complex.

Since the predictability property has traditionally outweighed the need for flexibility in systems design, the MAS technology has not yet succeeded in achieving its full potential. Despite a vibrant research activity, the industrial impact of MAS has been relatively low. However, the requirements on flexibility, autonomy, and openness of systems and their constituents are increasing. The question of whether and under which circumstances the MAS technology would be of practical interest was addressed in a position paper on the usefulness of MAS in power engineering (McArthur, et al. 2007). Although that paper was directed towards a specific application domain, most of the conclusions drawn there are of a rather general nature. The general recommendation is to use MAS in applications that would benefit tangibly from the autonomy of its constituent systems. Characteristics of such applications include:

- Requirements for interaction between distinct conceptual entities, such as different control subsystems and plant items. Potential benefits include simpler logic in each control entity, better track of local states, and increased ability to rapidly respond to unexpected events.
- A very large number of entities must interact, where it would be impossible to explicitly model the overall system behavior.
- There is enough data/information available locally to undertake an analysis/decision without the need for communication with a central point.
- New functions need to be implemented within existing legacy systems.
- Over time, there is a requirement for functionality to be continually added or extended.

When it comes to the benefits of MAS, (McArthur, et al. 2007) mention robustness, extensibility, flexibility, and improved modeling opportunities. The robustness stems partly from the idea of intelligent self-adaptation of actions to changing environment and partly from the relative simplicity of building in redundancy into MAS, both physical and functional, providing higher fault tolerance. Extensibility follows naturally from the encapsulating nature of agents (for example, legacy code wrapping is often mentioned as a particularly suitable application of MAS technology), while flexibility can be highlighted by the freedom to choose, either manually or automatically, an appropriate mix of agents in any given situation. Finally, the MAS architecture is particularly suitable for modeling and simulation of systems consisting of rather simple components that combined together produce complex emergent behavior.

To succeed in transferring the theoretical advancements within the MAS field to industrial applications, some challenges were identified:

- Common standards for data, platforms, and agent communication languages.

- Toolkits for the re-use of existing agent behaviors and capabilities.
- Security and measures to determine the level of trust between agents.
- Mobility of code.
- Lack of experience in the use of MAS technology in industry.

Additional challenges, complementary to the above list can be found in (Jennings, Sycara and Wooldridge 1998):

- Coordination of common goals:
  - How to formulate, describe, communicate and allocate problems and synthesize results among a group of intelligent agents?
  - How to enable individual agents to represent and reason about the actions, plans, and knowledge of other agents in order to coordinate with them? How to reason about the state of their coordinated process (e.g., initiation and completion)?
- Resource allocation:
  - How to effectively balance local computation and communication? More generally, how to manage allocation of limited resources?
- Conflicting goals and actions:
  - How to ensure that agents act coherently in making decisions or taking action, accommodating the nonlocal effects of local decisions and avoiding harmful interactions?
  - How to recognize and reconcile disparate viewpoints and conflicting intentions among a collection of agents trying to coordinate their actions?
  - How to avoid or mitigate harmful overall system behavior, such as chaotic or oscillatory behavior that may occur in a society of self-interested agents?

An additional complementary and somewhat overlapping list of MAS challenges is drawn by the roadmap for the agent-based technology, funded by the European Commission's Sixth Framework Programme (Luck, et al. 2005):

- Industrial strength software development methodology.
- Agreed standards for open systems development – agent- and non-agent based.
- Infrastructure for open agent communities – for example new web standards.
- Reasoning in open environments – e.g. automation of coalition formation, representation of norms and legislation, negotiation mechanisms, etc.
- Agent adaptation and learning – scalability and user trust are important issues.
- Trust and reputation – e.g. reputation mechanisms, social rules, enforcement of sanctions, electronic contracts, etc.

In addition to the above challenges, a speculative thought is that a broad introduction of MAS into industrial products has been hampered in some sense by the agents becoming too smart. In fact, the idea of cooperating agents was early adopted by the artificial intelligence (AI) society, to the extent that the agent technology has become the most active area of research within that field. Classical AI approaches to agent modeling were based on so-called deliberative or symbolic architectures (Wooldridge and Jennings 1995). The idea was to let each agent contain an explicit symbolic representation of the surrounding world, striving to attain human-like decision making via logical

reasoning, pattern matching and symbolic manipulation. The problem with this approach was that even rather trivial problems, modeled in this way, resisted efficient treatment. Extensions of symbolic representations to include beliefs, desires, intentions, time, etc. showed to lead to highly undecidable logic, rendering such approaches unusable in any time-constrained system (Chapman 1987) (Russell and Wefald 1991). Even successful solutions had difficulties to scale up.

Dark as the above picture might seem, it only represents a portion of the MAS history. In fact, the above shortcomings are mainly caused by an exaggerated focus on the AI aspects. However, the equality sign between MAS and AI is a common misconception. Instead, as pointed out in (McArthur, et al. 2007), MAS provide a framework for building distributed systems that may integrate different AI techniques if and where appropriate. Similar view is supported in (Wooldridge and Jennings 1998), where it is recommended to use a minimum of AI techniques to avoid one of the typical pitfalls of over-engineering the agents design.

At the same time, the AI field itself has experienced a significant progress, generating a large number of successful methods and commercial products. For example, to address the shortcomings of deliberative modeling, reactive architectures that did not use symbolic reasoning, were proposed. One of these was the subsumption architecture (Brooks 1991), arguing that complex systems should be built up incrementally into stable intermediate forms, based on a collection of simple rules instead of a complete representation of the world. Even though this approach needs to be tailor-cut for any specific application, it laid ground for the successful deployment of the Mars Pathfinder in 1997. Later, hybrid architectures that combine the best ideas of deliberative and reactive modeling have appeared, see e.g. (Wooldridge and Jennings 1995). Another area where agent-based technology has been successfully applied is the programming of wireless sensor networks (Fok, Roman and Lu 2005).

Of course, there is an ongoing activity to address the above challenges. For example, the already mentioned work on the usage of MAS in power engineering (McArthur, et al. 2007) identified nearly 70 MAS application papers, only within the power systems domain, that could be attributed to four main categories: distributed control, modeling and simulation, monitoring and diagnostics, and protection. There is also a unifying thrust in the MAS research, mainly in the form of the Foundation for Intelligent Physical Agents (FIPA), a standards organization that was formed in 1996 and included into IEEE Computer Society in 2005. Up to date, FIPA has produced a number of standards and agent communication languages (ACL), notably FIPA-ACL, that are gaining ground in the MAS community. This has laid the foundation for several open-source FIPA-compliant development toolkits for agent systems, such as JADE (JADE - Java Agent DEvelopment Framework 2011), Zeus (Zeus Agent Toolkit 2011), and FIPA-OS (FIPA-OS Toolkit 2011).

3 Conclusions

This review focused on five major areas of research that were identified to be related to the notion of federated embedded systems (FES), namely systems of systems (SoS), cyber-physical systems (CPS), pervasive / ubiquitous computing (Ubicomp), internet of things (IoT), and multi-agent systems (MAS). These research fields have been developing separately, both with respect to activity periods, research communities, and underlying theoretical bases.

While the SoS research sprung out of the engineering management field, it offers a taxonomy, suitable for different types of systems. Besides providing various SoS stakeholders with a common language, of course assuming that some SoS definition is adopted, the SoS research has addressed such issues as management of evolving systems, modeling, architecting, emergent behavior of complex systems, etc. The scope of possible applications is rather wide, including such domains as avionics, defense, wireless sensor networks, electrical power plants, robotics, transportation systems, health care, etc., see e.g. (Jamshidi 2008). The distributed, decentralized, heterogeneous, large-scale, and emergent nature of a typical SoS clearly resembles the idea of FES, especially if the socio-technical approach is adopted. An arguable difference is the stronger focus that the FES places on the technical side of the systems study, specifically targeting embedded systems.

On the other hand, the CPS term enjoys a coherent definition thanks to the excellent work on CPS foundations, carried out in 2006-2008. Despite this fact, the theoretical and technological focuses within the field are varying. As already mentioned, CPS related applications can be found in a number of different domains. When it comes to the diversity of the theoretical background, the above list of challenges, based in such research areas as real-time computing and networked control theory, speaks on its own. Partly due to the origins of the CPS field, currently quite a large portion of the ongoing work is concerned with timing or control issues. Differently from SoS or FES, the CPS community generally (but not always) seems to assume that the systems have a fix set of pre-engineered requirements and interconnections. The approach is strongly technical, as opposed to the more people and organization oriented focus of SoS. In our opinion, the focus of FES, including such aspects as business models, human-machine interaction, and open, dynamic system structures, fits in somewhere between the CPS and SoS fields in this respect.

The ubiquitous computing research seems to be a rather human centered discipline. In fact, the notion of invisibility, so prevalent in this field, presupposes that the human perception of pervasive systems is taken into consideration. This might explain why the majority of pervasive applications have traditionally been of a rather modest scale, focusing on limited groups of people that interact with the smart objects in their vicinity (Estrin, et al. 2002). However, this focus is starting to shift, with an increasing number of Ubicomp researchers realizing that the future belongs to large-scale heterogeneous systems (Lukowicz, Choudhury and Gellersen 2011).

The emerging research on the internet of things is strongly colored by the identification, communication and semantics technologies. It is cross-cutting between different disciplines, and incorporates or is interrelated with, either explicitly or implicitly, such fields as ubiquitous computing, ambient intelligence, semantic web technology, RFID, wireless sensor networks, and even multi-agent systems and cyber-physical systems. A difference with many other approaches to distributed systems is the scale of the envisioned global network. However, the actual network scale could be easily adaptable to any particular application, for example by means of secure gateways. Of course, such a solution would put high demands on the security and privacy technologies. Nonetheless, the prospects look promising, not least since several important funding strategy documents have recently selected the IoT as one of the core future challenges of our society.

Finally, a review of MAS-related literature draws a picture of a vibrant research field that has experienced its highs and lows during almost three decades. The numerous theoretical contributions are often software-centered and include such notions as software modeling and architecting, trust

and reputation, organizational and environmental structures, interoperability, and different kinds of intelligence aspects. A number of pitfalls, expectations, challenges, and applications have been examined and the ground seems to be paved for a more concentrated effort on transferring the research ideas into the industrial use. To restate the view of the power engineering community (McArthur, et al. 2007): *"MAS technology is maturing to the point where meaningful industrial applications are achievable."* The main missing ingredients are, at least to begin with, more agreement on standards, e.g. such as FIPA standards, agent development tools and methods, as well as more experience on industrial-scale implementations of the agent technology.

Apparently, the reviewed research fields overlap in several ways. Obvious similarities are their visions of smart, identifiable, heterogeneous, ubiquitous systems, often embedded in some sort of physical devices, which interact with each other and in consequence form higher level systems. The scales of such supra-systems vary, from the world-wide internet of things to relatively small systems of pervasive computers or multi-agent systems. Interestingly, while voices have been raised within the IoT-community about the need of studying smaller systems that are only connected to the IoT through some dedicated gateways, on the other side of the scales, both Ubicomp and MAS researchers realize the importance of scaling up their technology to the envisioned sizes of future networks of communicating objects. Similar reasoning can be recognized in the SoS community, represented by the idea of ultra-large-scale systems (Northrop, et al. 2006).

A look at the applications that are addressed by the above research directions also reveals a strong coherence. Instead, the differences between the fields lie mainly in their focuses of study and scientific backgrounds. It seems thus quite believable that the above fields are about to merge to some extent. In the process, challenges within all of the above focus fields will require to be addressed, such as for example systems architecture, process management, scheduling, reliable communication, identifiability, human-machine interfaces, smartness, flexibility, to mention just a few. Hopefully, the anticipated merging will roll out consciously, drawing on the unique progresses and competences of each of the highlighted fields.

4 Bibliography

Abowd, G. D., and E. D Mynatt. "Charting Past, Present, and Future Research in Ubiquitous Computing." *ACM Transactions on Computer-Human Interaction* 7 (2000): pp. 29-58.

Abowd, G., A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles. "Towards a Better Understanding of Context and Context-Awareness." In *Handheld and Ubiquitous Computing*, edited by H.-W. Gellersen, pp. 304-307. Berlin: Springer, 1999.

Ackoff, Russell L. "Towards a System of Systems Concepts." *Management Science* 17 (1971): pp. 661-671.

Ashton, K. "That 'Internet of Things' Thing." *RFID Journal*, 2009.

Atzori, L., A. Iera, and G. Morabito. "The Internet of Things: a Survey." *Computer Networks*, 2010.

Baldauf, M., S. Dustdar, och F. Rosenberg. ”A Survey on Context-Aware Systems.” *Int. J. Ad Hoc and Ubiquitous Computing* 2 (2007): pp. 263-277.

Bardram, J., and A. Friday. "Ubiquitous Computing Systems." In *Ubiquitous Computing Fundamentals*, edited by J. Krumm. Redmond, Washington: CRC Press, 2009.

Boardman, J., and B. Sauser. "System of Systems - the Meaning of Of." *International Conference on System of Systems Engineering.* Los Angeles: IEEE, 2006. pp. 118-123.

Boardman, J., S. Pallas, B. J. Sauser, and D. Verma. *Report on System of Systems Engineering.* Final Report for the Office of Secretary of Defense, Hoboken, NJ: Stevens Institute of Technology, 2006.

Brooks, R. A. "Intelligence without Representation." *Artificial Intelligence* 47 (1991): pp. 139-159.

Carlock, P. G., and R. E. Fenton. "System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations." *Systems Engineering* 4 (2001): pp. 242-261.

Chapman, D. "Planning for Conjunctive Goals." *Artificial Intelligence* 32 (1987): pp. 333-378.

Chen, G., and D. Kotz. *A Survey of Context-Aware Mobile Computing Research.* tech. report, Department of Computer Science, Dartmouth College, 2000.

Cluster of European Research Projects on the Internet of Things. *Vision and Challenges for Realising the Internet of Things.* Edited by H. Sundmaeker, P. Guillemin, P. Friess and S. Woelfflé. Brussels: European Commision - Information Society and Media, 2010.

CPS Steering Group. *Cyber-Physical Systems: Executive Summary.* US National Workshop on Cyber-Physical Systems, 2008.

CPS Summit Report. 2008. http://varma.ece.cmu.edu/Summit/.

DeLaurentis, D. "Understanding Transportation as a System of Systems Design Problem." *AIAA Aerospace Sciences Meeting.* Reno, NV, 2005.

DeLaurentis, D., and R. K. Callaway. "A System-of-Systems Perspective for Public Policy Decisions." *Review of Policy Research* 21 (2004): pp. 829-837.

Dunkels, A., B. Grönvall, and T. Voigt. "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors." *International Conference on Local Computer Networks.* Washington: IEEE, 2004. pp. 455-462.

Estrin, D., D. Culler, K. Pister, and G. Sukhatme. "Connecting the Physical World with Pervasive Networks." *IEEE Pervasive Computing*, 2002: pp. 59-69.

FIPA-OS Toolkit. 2011. http://fipa-os.sourceforge.net/index.htm.

Fok, C.-L., G.-C. Roman, and C. Lu. "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications." *Conference on Distributed Computing Systems.* Columbus: IEEE, 2005. pp. 653-662.

Fovino, Igor Nai, and Marcelo Masera. "Emergent Disservices in Interdependent Systems and Systems-of-Systems." *IEEE International Conference on Systems, Man and Cybernetics.* Taipei, 2006. pp. 590-595.

Franklin, S., and A. Graesser. "Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents." *Lecture Notes in Computer Science* 1193 (1997): pp. 21-35.

Gorod, A., B. Sauser, and J. Boardman. "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward." *IEEE Systems Journal* 4 (2008): pp. 484-499.

Guinard, D., and V. Trifa. "Towards the Web of Things: Web Mashups for Embedded Devices." *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009).* Madrid, 2009.

Guinard, D., V. Trifa, F. Mattern, and E. Wilde. "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices." In *Architecting the Internet of Things*, edited by D. Uckelmann, M. Harrison and F Michahelles, pp. 97-129. Berlin: Springer, 2011.

Haller, S., S. Karnouskos, and C. Schroth. "The Internet of Things in an Enterprise Context." In *Future Internet - FIS 2008*, edited by J. Domingue, D. Fensel and P. Traverso, pp.14-28. Berlin: Springer, 2009.

Hitchins, D. *Advanced Systems Thinking, Engineering, and Management.* Norwood: Artech House, 2003.

JADE - Java Agent DEvelopment Framework. 2011. http://jade.tilab.com/.

Jamshidi, Mo, ed. *System of systems engineering, Innovations for the 21st Century.* The Wiley and Sons, 2008.

Jamshidi, Mo, ed. *Systems of systems engineering: Principles and applications.* CRC Press, 2008.

Jennings, N. R. "On Agent-Based Software Engineering." *Artificial Intelligence* 117 (2000): pp. 277-296.

Jennings, N. R., K. Sycara, and M. Wooldridge. "A Roadmap of Agent Research and Development." *Autonomous Agents and Multi-Agent Systems* 1 (1998): pp. 275-306.

Karl, H., och A. Willig. *Protocols and Architectures for Wireless Sensor Networks.* Wiley, 2005.

Katasonov, A., O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan. "Smart Semantic Middleware for the Internet of Things." *International Conference on Informatics in Control Automation and Robotics.* Funchal, Madeira, 2008. pp. 169-178.

Keating, C., et al. "System of systems engineering." *IEEE Engineering Management Review* 36 (2003): pp. 62-62.

Keating, C.B. "Research foundations for system of systems engineering." *IEEE International Conference on Man, Systems and Cybernetics.* 2005. pp. 2720-2725.

Kotov, V. *Systems-of-Systems as Communicating Structures.* Hewlett-Packard Company, 1997.

Krygiel, A. J. *Behind the Wizard's Curtain: an Integration Environment for a System of Systems.* Vienna: CCRP Publication Series, 1999.

Lane, J. A., and R. Valerdi. "Synthesizing SoS Concepts for Use in Cost Estimation." *International Conference on Systems, Man and Cybernetics.* Los Angeles: IEEE, 2005. pp. 993-998.

Lee, E.A. *Computing Needs Time.* Tech. Rep., EECS Department, University of California, Berkeley, 2009.

—. "Cyber Physical Systems: Design Challenges." *Symposium on Object Oriented Real-Time Distributed Computing (ISORC).* IEEE Computer Society, 2008. pp. 363-369.

Levis, P., et al. "Tinyos: An operating system for sensor networks." In *Ambient Intelligence*, edited by W. Weber, J. M. Rabaey and E. Aarts, pp. 115-148. Springer Verlag, 2005.

Luck, M., P. McBurney, O. Shehory, and S. Willmott. *Agent Technology: Computing as Interaction, a Roadmap for Agent Based Computing.* AgentLink, 2005.

Lukasik, S. J. "Systems, Systems of Systems, and the Education of Engineers." *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 12 (1998): pp. 55-60.

Lukowicz, P., T. Choudhury, and H. Gellersen. "Beyond Context Awareness." *IEEE Pervasive Computing* 10 (2011): pp. 15-17.

Lyytinen, K., and Y. Yoo. "Issues and Challenges in Ubiquitous Computing." *Communications of the ACM* 45 (2002): pp. 62-65.

Maier, Mark W. "Architecting Principles for Systems-of-Systems." *Systems Engineering* 1 (1998): pp. 267-284.

Manthorpe Jr., W. H. J. "The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL." *Johns Hopkins APL Technical Digest* 17 (1996): pp. 305-310.

McArthur, S. D. J., et al. "Multi-Agent Systems for Power Engineering Applications - Part 1: Concepts, Approaches, and Technical Challenges." *IEEE Transactions on Power Systems* 22 (2007): pp. 1743-1752.

Northrop, L., et al. *Ultra-Large-Scale Systems: The Software Challenge of the Future.* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.

Rajkumar, R., I. Lee, L. Sha, and J. Stankovic. "Cyber-Physical Systems: the Next Computing Revolution." *the 47th Design Automation Conference.* New York, NY: ACM, 2010. pp. 731-736.

Russell, S. J., and E. Wefald. *Do the Right Thing - Studies in Limited Rationality.* Cambridge: MIT Press, 1991.

Russell, S. J., and P. Norvig. *Artificial Intelligence: a Modern Approach.* Englewood Cliffs, NJ: Prentice Hall, 1995.

Sage, Andrew P., and Christopher D. Cuppan. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems." *Information-Knowledge-Systems Management* 2 (2001): pp. 325-345.

Sage, Andrew P., and William B. Rouse. *Handbook of Systems Engineering and Management.* John Wiley and Sons, 1999.

Sage, Andrew P., och William B. Rouse. *Handbook of Systems Engineering and Management.* John Wiley and Sons, 1999.

Saha, D., and A. Mukherjee. "Pervasive Computing: a Paradigm for the 21st Century." *IEEE Computer Society*, 2003: pp. 25-31.

Sarma, S., D.L. Brock, and K. Ashton. *The Networked Physical World, Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification.* White paper, Cambridge: Auto-ID Center, 2000.

Satyanarayanan, M. "Pervasive Computing: Vision and Challenges." *IEEE Personal Communications*, 2001: pp. 10-17.

Schilit, B. N., N. I Adams, and R. Want. "Context-Aware Computing Applications." *Workshop on Mobile Computing Systems and Applications.* Santa Cruz: IEEE Computer Society, 1994. pp. 85-90.

Sha, L., S. Gopalakrishnan, X. Liu, and Q. Wang. "Cyber-Physical Systems: A New Frontier." *MACHINE LEARNING IN CYBER TRUST* 1 (2009): pp. 3-13.

Shenhar, A.J. "One Size Does Not Fit All Projects: Exploring Classical Contingency Domains." *Management Science*, 2001: pp. 394-414.

Sousa-Poza, Andres, Samuel Kovacic, and Charles Keating. "System of systems engineering: an emerging multidiscipline." *International Journal of System of Systems Engineering* 1 (2008): pp. 1-17.

US President's Council of Advisors on Science and Technology (PCAST). "Federal Networking and Information Technology R&D (NITRD) Program Review." Advisory comittee report, Washington, 2007.

Weiser, Mark. "The Computer for the 21st Century." *Scientific American*, 1991: pp. 94-104.

Wooldridge, M. J., and N. R. Jennings. "Pitfalls of Agent-Oriented Development." *Second International Conference on Autonomous Agents.* New York: ACM, 1998. pp. 385-391.

Wooldridge, M., and N. R. Jennings. "Intelligent Agents: Theory and Practice." *Knowledge Engineering Review* 10 (1995): pp. 115-152.

Ye, W., J. Hedemann, and D. Estrin. "An Energy-Efficient MAC Protocol for Wireless Sensor Networks." *International Conference on Computer Communications.* New York: IEEE, 2002. pp. 1567-1576.

Yick, J., B. Mukherjee, and D. Ghosal. "Wireless sensor network survey." *Computer Networks*, 2008: pp. 2292-2330.

Zeus Agent Toolkit. 2011. http://sourceforge.net/projects/zeusagent/.