FAULT TOLERANT SENSING MODEL FOR CYBER-PHYSICAL SYSTEMS

Sudeep Ghimire
CTS, UNINOVA, FCT, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
sud@uninova.pt

João Sarraipa
CTS, UNINOVA, FCT, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
jfss@uninova.pt

Carlos Agostinho
CTS, UNINOVA, FCT, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
ca@uninova.pt

Ricardo Jardim-Goncalves
CTS, UNINOVA, FCT, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
rg@uninova.pt

ABSTRACT

A sensing system is capable of making decisions by using information captured through physical and virtual objects and providing value added information to enhance its global context awareness. Sensing system forms one of the core components of cyber-physical systems (CPSs) which provide the interface for data collection from physical world and detecting various situations. Thus, it becomes critical to develop sensing model that is fault tolerant towards the failure of components such as sensors and actuators. This will enable realization of CPSs that will continue to run even when the interface to the physical world fails. This research work presents the model for defining sensing model for representing sensors and actuators and creating virtual objects that will enable data regeneration when the physical devices fails. The data regeneration algorithm is based on the virtual instance attributed with contextual details of the deployed physical devices. Fault tolerant systems in an important research area for CPSs and this paper addresses this challenge to enable development of robust and smart CPSs.

Keywords: Cyber-physical Systems, Sensing Systems, Fault Tolerance, Data Regeneration.

1 INTRODUCTION

In the recent time, there has been a rapid research and technological advances in embedded systems supported by developments in wireless communications and increasing availability of sensors, actuators, and mobile devices. This has led to new ubiquitous computing paradigm that facilitates computing and communication services anytime, everywhere providing value added services over real-time data. This emerging paradigm is changing the way we live and work today and giving rise to more complex systems. Global and localized networks, users, sensors, devices, systems and applications can seamlessly interact with each other and even the physical world in unprecedented ways. This clearly depicts the development of "systems-of-systems" that interact with real-world environment or of "systems" that have equally close connection with both the physical and the computational components. The new advancement in technology opens up the need for systems to provide seamless integration of physical world with the digital world. Such systems are termed "Cyber-Physical Systems" (CPS), which act independently, co-operatively or as "systems-of-systems" composed of interconnected autonomous systems originally independently developed to fulfil dedicated tasks. In general, CPSs refer to the next generation of engineered systems that require tight integration of computing, communication, and control technologies

SpringSim-Mod4Sim 2017, April 23-26, Virginia Beach, VA, USA

to achieve stability, performance, reliability, robustness, and efficiency in dealing with physical systems of many application domains (Guturu and Bhargava 2011).

1.1 Cyber-Physical Systems

The growing trend toward computational intelligence, automation, and control for complicated but well-defined tasks or processes, especially when demands or constraints are not amenable to human intervention paves the path for application of CPS. For example, automatic collision systems could detect moving objects and respond faster than a human operator. Unmanned CPS could be used to reduce the risk to human life by detecting mines, exploring volcanoes, or conducting otherwise hazardous tasks. Machines driven by a computer do not suffer fatigue and may be more precise than is humanly possible. In future CPS could make possible concepts only imagined today, such as unmanned tours to the moon, bionic suits, and automated largescale indoor agriculture systems (NIST 2013).

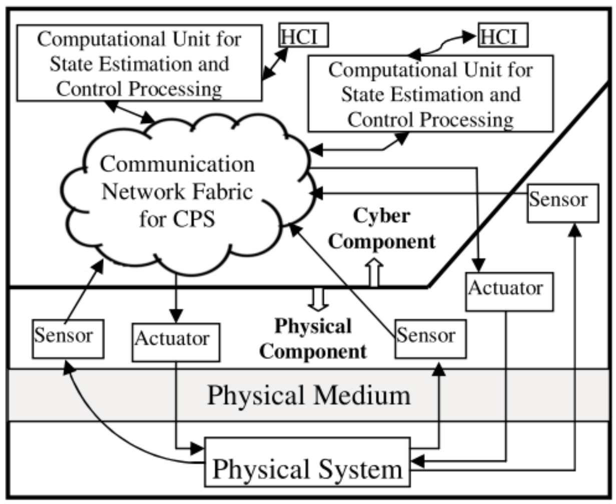

Figure 1: Nuts and Bolts of CPS.

The physical component of CPS is composed of physical devices with capabilities such as integrated networking, information processing, sensing and actuation, can operate in close relation with the physical environments as depicted in **Error! Reference source not found.** taken from (Guturu and Bhargava 2011). These abilities can lead towards the realization of systems that can be responsive to real-time changes in the environment. While, the cyber component comprises of computational unites that has proven capability for high end computation, storage and analytics. Such tightly coupled cyber and physical systems that exhibit high level of integrated cooperation and intelligence are characteristics of CPS.

1.2 Sensing System Engineering for CPS

Sensing system is capable of making decisions by using information captured through physical and virtual objects and providing value added information to enhance its global context awareness. In fact we have generalized the term sensing enterprise as defined by FInES Cluster (FInES Cluster 2011). The sensing system combines the concepts of sensors with mobile technology and distributed intelligence to perform analysis and decision-making, both in the real and digital worlds. This concept is a cornerstone for enterprise level CPS systems and is supported by the anticipation that sensors will become a commodity in future (Santucci, Martinez, and Vlad-câlcic 2012) and (Berger et al. 2016). Note that sensing system will not only be dependent on physical sensors but also on virtual sensors i.e. enterprise will not have physical access to the sensor but can have access to the observations of these sensors, which is also highlighted in (Santucci, Martinez, and Vlad-câlcic 2012).

Sensors and actuators form the basic entities of sensing system of CPS. Sensors and actuators immersed in the physical environment are connected to computing nodes, which are often single-board computers like Raspberry PI or BeagleBoard. However, as the diversity of device protocols and their properties rises, it requires a unified solution for implementing and integration of such devices into the cyber component of CPS. Thus, one of the most important properties to be addressed, for sensing system engineering for CPS is the methodology for virtualization of physical resources so that they can be shared and utilized by different entities in the overall CPS ecology. In general, virtualization refers to process involved in converting a physical view to a logical view. Virtualization serves two important purposes: it creates a simpler abstract view of physical resource for consumers and it creates the possibility for re-combination of functionalities of physical resources to create complex logical resources. In fact, one of the most common differences between the physical and logical views is that a physical resource may be shared by several processes, but logically, the processes are unaware of each other. Each process can have its own logical resource, even though there may only be a single physical resource. This has also been studied in (Shafiq et al. 2015) from industrial point of view. In general, virtualization will allow the physical devices to be represented as virtual objects (VOs). The VO concept aims to hide the heterogeneity of the physical devices, in terms of capabilities, functionalities and communication protocols. At the same time, VOs will offer a transparent and ubiquitous access to the physical devices via a well-defined interface, and allow agents to connect directly without needing to consider proprietary drivers or needing to address fine-grained technological issues. VOs for sensorial notes are also discussed in some other works such as (Lichuan Liu, Kuo, and Zhou 2009; Lin et al. 2007) and is used to provide feasible and economical alternatives to costly or impractical physical measurement instrument, by utilizing the functionalities of existing physical sensor. In complex CPS there is the need to for the use of in-network data processing techniques (Fouad et al. 2015).

1.3 Fault Tolerant System

Fault tolerance can be defined as the property that enables any system to continue operating properly in the event of the failure of some of its components. Fault tolerance is particularly sought after in critical systems, thus making an important characteristic for CPS, where it is not accepted that the failure in some components can cause total system breakdown. The most sought after feature for CPS is the ability of maintaining system functionality when portions of a system break down which is referred to as graceful degradation (Gonzalez et al. 1997). In general, such systems can be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state. For the case of sensing system of CPS the fault under consideration in the scope of this research work is the failure of sensor and actuators thus creating the void for data collection and action propagation. Thus, the anticipated failure conditions for sensors is missing data while for actuators is failure of performing action. The solution proposed in this research work for fault mitigation is data/action regeneration for the components that have failed at run-time.

However, it is important to state that the solution formulated in the scope of this research paper is not for the case when the system failures are catastrophic, or the cost of making it sufficiently reliable is very high. In such cases, the opted solution will be to have functionality for rollback recovery and can be a human action if humans are present in the loop. In addition, during the system design time providing fault-tolerant design for every component is normally not an option because that can increase in system weight, size, power consumption, cost, as well as time to design, verify, and test. Therefore, a number of choices have to be examined to determine which components should be fault tolerant, which are quite interestingly discussed in (Dubrova 2013).

2 SENSORIAL MODEL

In order to support a complex ubiquitous computing environment, sensing model will need to support localized cooperation of sensor and actuator nodes to perform complicated application directed tasks and in-network processing. Data collected from sensors needs to be transformed into high-level abstract information, which is not necessarily a measurement the physical sensors themselves can provide. At the same time, the actions generated by different business processes needs to be transformed into actions that can be actuated via the actuator notes. Therefore, it is important to define formal abstraction for modelling the sensorial system of the CPS system. In a generic term, sensors are devices that are capable of reading physical and physiological parameters while actuators are physical devices that are capable of performing actions based on the control statements provided to them. In the following subsections, we will provide the formal model for representation of sensors and actuators that will form the core for the realization of virtual objects.

2.1 Model Formulation

In this section, we will provide a formal model of sensors, actuators and VOs, which will be used for modelling the overall sensing system.

A sensor *S* can be defined by a tuple $< D, C, C' >$ where

- *D* represents data-stream and is defined as *D*=$< DataType, \{R_1 \dots R_n\} >$ s.t.
  - *DataType* defines the datatype of the data that is collected by the sensor and
  - $\{R_1 \dots R_n\}$ is an ordered set of readings obtained at various time s.t. for a time instance k the event $A_{ex}(k)$ to be detected from the collected data is given by $A_{ex}(k) = f(R_{m-n})$, where function f is used to generate event from the sensor reading at time instance k by considering data *R* between time instances *m* and *n*.
- *C* provides the metadata of the sensor *S* and is used for understanding the contextual use of the sensor and can have a URI to an external sensor networks ontology or can also be linked with other instances of metadata resources.
- *C'* defines the configuration of the sensor *S*, which includes parameters that define the working configurations of the sensor such as protocols, i/o data ranges, i/o interfaces etc.

Based on this definition sensors comprising the sensing system can be modelled. After defining the sensors, we need to define actuators. Raw definition of an actuator is - mechanism that puts something into automatic action. While, from engineering point of view, actuators are a subdivision of transducers and they are devices that transform an input signal (mainly an electrical signal) into some form of motion.

An actuator *A* can be defined by a tuple $< Act, C, C' >$ where

- C and $C'$are same as defined for sensors.
- Act represent a sequence of actions i.e. is an ordered set represented as $\{Act_1 \dots Act_n\}$, an ordered set of actions to be undertaken by the actuator at various time s.t. for a time instance k, $Act_1 = f'(A_{out}(k))$ where function f' is used to generate action for actuator from the output action of the situational controller.

Based on this definition individual sensors and actuators comprising a sensing system can be modelled. However, to model a complex CPS, we need to define virtual sensor (VS) and virtual actuators (VA). A virtual sensor uses information available from other measurements and process parameters to calculate an estimate of the quantity of interest. VSs are termed virtual because they do not have physical occurrence but are formed by logical infusion of various sensors. Moreover, the similar definition for VA is applicable with the only difference being the control action for actuators rather than the data inputs for sensors.

A virtual sensor *VS* can be defined by a tuple $< S, O, D, C, C' >$ where

- *S* is set of data sources such that $S = \{S_1, \dots, S_n\} \cup \{VS_1, \dots, VS_m\}$ where S_i is physical sensor and VS_i is Virtual sensor that has been define before. Thus, existing VS can be used for creation of new VS.
- *O* is the operator relationship between sensors *S*. This operator is used to aggregate the data for VS based on the data collected from the constituent sensors.

- D is data-stream and is defined as $D=<DataType, \{R_1 \ldots R_n\}>$, exactly as in definition of sensors. The only difference is the reading R_n is computed by applying the operator O on the readings R of sensors S by keeping the order hence time intact.
- C and C' are defined similarly as for sensors.

Thus, the most important part of VS is the operator O. The relation O can have different syntax and semantics as needed based on the type of data infusion to be established between sensors. For instance if VS is used for taking into consideration humidity and temperature of a body part (e.g. arm, armpit) at the same time then a one to one union over the reading of the sensors for humidity and temperature will give an ordered set of reading for this new VS. Moreover, if the operator is to find the difference of the reading of two sensors (e.g. environment and body temperature) then the ordered difference between each element of the reading of two sensors will give the dataset for the VS. Note that the context of the VS can always be obtained by the union of the individual contexts of the sensors forming the VS.

A virtual actuator VA can be defined by a tuple $< A, O, Act, C, C' >$ where

- A is set of actuators such that $A = \{A_1, \ldots, A_n\} \cup \{VA_1, \ldots, VA_m\}$ where A_i is physical actuators and VA_i is virtual actuators sensor that has been define before. Thus, existing VA can be used for creation of new VA.
- O is the operator relationship between actuators A. This operator is used to aggregate the actions for VA based on the actions generated by the business processes.
- Act is set of actions generated for actuator(s) and is defined as $\{Act_1 \ldots Act_n\}$, exactly as in definition of actuators. The only difference is the action Act_n is computed by applying the operator O on the actions Act of actuators related by the operator O.
- C and C' are defined exactly as the same of actuators.

Thus, this definition of VA can be used for representation of complex actuators that can take care of multiple actions for single event. For a simple example let us say a fire is detected and different actions needs to be triggered like fire alarm, water sprinkle, notifications etc. The process that detects the fire can trigger only one VA, which is actually composed of three actuators and performs three actions in the physical space. This helps the abstraction of the physical actuators thus making the CPS modelling independent of the actual actuator devices available.

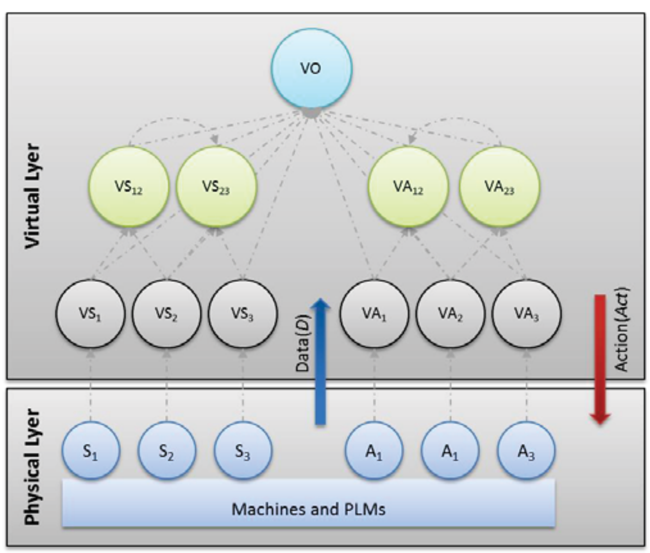

Figure 2: High level view of the sensing model.

Based on the presented model the overall components of sensing system can be represented as shown in Figure 2. In the figure, note that the dotted grey arrows represents the logical relationships between components at physical and virtual layers, which is not to be confused with the data and control flow. Hence, virtual objects (VOs) of the sensorial model are composed of virtual sensors and virtual actuators, which can be defined as $VO = \{VS \cup VA\}$.

2.2 Data Regeneration on Device Failure

In order to define fault tolerance model, we would present the solution for data regeneration in case of device node failure. In the scope of this work, it is termed as *Reinforcement Sensing (RS)* with the main idea of providing the action or process of reinforcing or strengthening failure node in reference to non-failure nodes of the overall system. *RS* is important for distributed data sensing that can deal with sensor nodes malfunctioning at run-time. *RS* is based on the principle of utilizing data and information processing by taking advantage of data dependencies and redundancies that may exist in sensorial network due to components sensing the same or similar property (refer (Teacy et al. 2009) and (Krohn et al. 2005) for further understanding on reinforcement sensing also termed as collaborative sensing by other authors). RS involves two computational steps:

- RS Analysis - identification of data dependencies and
- RS Approximation - approximation of value for property P.

The RS Approximation is based on the dependency relations of P_i and thus can be calculated by using the similarity between the properties and associated concepts, which is calculated in the process of RS analysis. While, the RS analysis is a more complex task and bears the computational overhead of data collection vs. the readiness of dependency relation. In order to perform both the tasks the model of sensors and/or virtual sensors as given before play the most important role. Note that in the following section the virtual instance of the sensor is considered and is termed as S for readability purpose.

Reinforcement Sensing between two sensors S_x and S_y can be defined by the tuple $< S_x, S_y, \varphi, \mu >$ where φ *and* μ are the functions used to calculate the similarity vector between the two sensors, φ for calculation of semantic distance between the sensor context and μ for the data similarity.

Let us first provide the details on the methodology for calculating semantic distance i.e. φ. In this process for similarity measurement, weight allocation is made between the concepts nodes representing the sensor model, more precisely C and C' as stated in definition of sensors and actuators are concepts from ontology that have weights allocated for all the relationships between concepts. For two concepts C_1 and C_2 which are related with the relation sub-class (*sub*), the weight allocation is calculated as:

$$w[sub(C_1, C_2)] = 1 + \frac{1}{k^{depth(C_2)}}$$

Where, *depth(C)* presents the depth of concept C from the root concept to node C in ontology hierarchy, k is a predefined factor larger than 1 indicating the rate at which the weight values decrease along the ontology hierarchy.

This formula has two important properties:

- The semantic differences between upper level concepts are higher than those between lower level concepts. In other words, two general concepts are less similar than two specialized ones.
- The distance between sibling concepts is greater than the distance between parent and child concepts. Specially, the depth of root concept is zero, and the depth of other concepts equal to their path length to root concept node.

In addition, if there are multiple inheritances relation between concepts (such as $C_1 \subset C_3, C_1 \subset C_2$), the depth of the concept node C_1 can have multiple values, and thus the weight will have multiple values. This process allows calculating the similarity between the sensors under consideration. If the distance is very high then the following step for data approximation can be ignored.

Now, in order to regenerate the data for sensor that failed which is semantically similar to some of the sensor that are working fine, let us assume that the sensor data manager collects the values of preselected data fields of the set of sensors in the latest time instances $t = 1. . n$. Furthermore, for acquiring the dependency relation, RS Analysis checks all the aggregated data to find out which data fields are dependent on others.

Let $S_x.d_j^t$ be the value of data field d_j of sensor S_i at a time instance t. Hence, $\{S_i.d_j\}_1^n$ denote the time series of the data field d_j of component S_i at time instances 1 to n. Now, let $\mu_{d_j}(S_x.d_j^t, S_y.d_j^t)$ be the distance between two data values of d_j^t in components S_x and S_y measured by metric μ specific to d_j^t. Then, for all component pairs, S_x , S_y $x \neq y$, having the fields d_m and d_n, RS Analysis computes the boundary Δ_{d_j} such that the implication $\mu_{d_j}(S_x.d_j^t, S_y.d_j^t) < \Delta_{d_j} \Rightarrow \mu_{d_m}(S_x.d_m^t, S_y.d_m^t) < T_{d_m}$ for the time instances $t = 1. . n$ is satisfied in (at least) the specified percentage of all the cases (confidence level a_d, e.g. 90%). Here T_{d_m} represents the tolerable distance threshold and is provided for each d_m. Based on this definition the RS Analysis concludes that the value of $S_x.d_j^t$ is close to the value of $S_y.d_j^t$ (and vice versa) for t such that the values of $S_x.d_l^t$ and $S_y.d_l^t$ are close as well. Thus, when a sensor S_i fails to sense the values of d_m, an approximation of this property has to take place. This is done by creating data stream with the exchange function $S_x.d_m := S_y.d_m$ and membership condition $\mu_{d_j}(S_x.d_j^t, S_y.d_j^t) < \Delta_{d_j}$. If more than one S_y satisfies the membership condition, an arbitrary one is selected. Thus, created data stream becomes the collected data for failed sensor.

Note that the task to compute the boundary Δ_{d_j} is resource and time demanding but some techniques that can be applied to lower the time such as sorting the data according to $\mu_{d_j}(S_x.d_j^t, S_y.d_j^t)$ or using sampling of the gathered data to obtain a statistically significant answer. Some other techniques such as linear regression, k-nearest neighbors, neural networks etc. can be applied to detect dependencies between data.

3 CONCLUSION

The sensorial system modelling is an important task for defining overall CPS based systems. Sensorial systems are important for situational modelling which are used for detection of different situations and providing necessary actions to address the detected situations. This will enable the CPS to have the perception of environmental elements and events occurring at different time and also space where necessary. In the presented model the parameters of sensors and actuators representing data (D_k) and action (Act_k) respectively can be related to the behavioural model of CPS which can be represented with hybrid automata. This model thus can form the base for formulation of methodology for automated verification and validation of CPS models. In the presented sensorial model it is important to note that each computational node that hosts VOs will also contain the agent server that will provide the functionality to directly exploit the physical part through VO abstraction. This is an important consideration so that less data needs to be transferred over a long distance (i.e. toward remote hosts) and local access and computation is fostered in order to achieve good performance and scalability. The data regeneration methodology presented in this research work will play an important role for implementation of sensorial systems that can tackle system failures in case of malfunctioning of some of the sensor nodes, which is a very interesting scenario for complex CPS based systems. The presented sensorial model needs to be enhanced with necessary semantic representation to enable formal model verification and validation and to check consistency and satisfiability conditions, which is one of the most important future works. Also, in future, we aim to provide a technical solution for realization of sensing model by utilizing semantics based meta-modelling languages like OWL and technical implementation of the overall framework for virtualization and data regeneration. We also aim to perform further experiments over the proposed model for reinforcement sensing by considering some industrial use-cases within the scope of the project virtual factory operating system (vf-OS).

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union H2020 Programs under grant agreements No 611312 vf-OS and No 636909 C2NET, as well as FCT - Fundação para a Ciência e Tecnologia, research unit CTS - Centro de Tecnologia e Sistemas - reference number UID/EEA/00066/2013.

REFERENCES

Berger, Christoph, Andreas Hees, Stefan Braunreuther, and Gunther Reinhart. 2016. "Characterization of Cyber-Physical Sensor Systems." *Procedia CIRP* 41. Elsevier B.V.: 638–43. doi:10.1016/j.procir.2015.12.019.

Dubrova, Elena. 2013. *Fault-Tolerant Design*. *Fault-Tolerant Design*. New York, NY: Springer New York. doi:10.1007/978-1-4614-2113-9.

FInES Cluster. 2011. "Position Paper on Orientations for FP8: A European Innovation Partnership for Catalysing the Competitiveness of European Enterprises." http://cordis.europa.eu/fp7/ict/enet/documents/fines-position-paper-fp8-orientations-final.pdf.

Fouad, Mohamed Mostafa, Nour E. Oweis, Tarek Gaber, Maamoun Ahmed, and Vaclav Snasel. 2015. "Data Mining and Fusion Techniques for WSNs as a Source of the Big Data." *Procedia Computer Science* 65 (Iccmit). Elsevier Masson SAS: 778–86. doi:10.1016/j.procs.2015.09.023.

Gonzalez, O., H. Shrikumar, J.A. Stankovic, and K. Ramamritham. 1997. "Adaptive Fault Tolerance and Graceful Degradation under Dynamic Hard Real-Time Scheduling." In *Proceedings Real-Time Systems Symposium*, 79–89. IEEE Comput. Soc. doi:10.1109/REAL.1997.641271.

Guturu, Parthasarathy, and Bharat Bhargava. 2011. "Cyber-Physical Systems : A Confluence of Cutting Edge Technological Streams." In *International Conference on Advances in Computing and Communication*. https://www.cs.purdue.edu/homes/bb/CPSReviewPaper.pdf.

Krohn, Albert, Tobias Zimmer, Michael Beigl, and Christian Decker. 2005. "Collaborative Sensing in a Retail Store Using Synchronous Distributed Jam Signalling." In *International Conference on Pervasive Computing (Pervasive) 2005*, 237–54. doi:10.1007/11428572_15.

Lichuan Liu, Sen M. Kuo, and MengChu Zhou. 2009. "Virtual Sensing Techniques and Their Applications." In *2009 International Conference on Networking, Sensing and Control*, 31–36. IEEE. doi:10.1109/ICNSC.2009.4919241.

Lin, Bao, Bodil Recke, Jørgen K.H. Knudsen, and Sten Bay Jørgensen. 2007. "A Systematic Approach for Soft Sensor Development." *Computers & Chemical Engineering* 31 (5–6): 419–25. doi:10.1016/j.compchemeng.2006.05.030.

NIST. 2013. "Strategic R&D Oportunities for 21st Century Cyber-Physical Systems." *Foundations for Innovation in Cyber-Physical Systems Workshop*. http://www.nist.gov/el/upload/12-Cyber-Physical-Systems020113_final.pdf.

Santucci, Gérald, Cristina Martinez, and Diana Vlad-câlcic. 2012. "The Sensing Enterprise." In *FInES Workshop at FIA 2012*, 1–14. http://www.theinternetofthings.eu/sites/default/files/%5Buser-name%5D/Sensing-enterprise.pdf.

Shafiq, Syed Imran, Cesar Sanin, Edward Szczerbicki, and Carlos Toro. 2015. "Virtual Engineering Object / Virtual Engineering Process: A Specialized Form of Cyber Physical System for Industrie 4.0." *Procedia Computer Science* 60 (1). Elsevier Masson SAS: 1146–55. doi:10.1016/j.procs.2015.08.166.

Teacy, W T L, J Nie, S McClean, G Parr, S Hailes, S Julier, N Trigoni, and S Cameron. 2009. "Collaborative Sensing by Unmanned Aerial Vehicles." *Communication*, 13–16. http://www.cs.ox.ac.uk/files/3063/atsn.pdf.

AUTHOR BIOGRAPHIES

Sudeep Ghimire received PhD degree in Industrial Information Systems from New University of Lisbon, and is currently working as researcher at Centre of Technology and Systems, FCT-UNL, Portugal. He has participated in several research and innovation projects specifically in the domain of factories of future with main contributions in formulation of technical solutions. His main fields of interest are Industrial Information Systems, Cyber-Physical Systems, Evolutionary Computation and Service-Oriented Systems. His email address is sud@uninova.pt.

João Sarraipa is currently a senior researcher at UNINOVA research center where he has been working since 2001. He holds a PhD in the area of industrial information systems of the Electrical and Computer Engineering by the Faculty of Science and Technology of the New University of Lisbon (FCT/UNL) and is deeply involved in the areas of Knowledge Management, Ontologies, Semantic Interoperability, and eTraining. He has experience in R&D and has participated in various projects like IDEAS (IST-2001-37368), INTEROP (IST-508011), CoSpaces (IST-034245) etc. and has contributed to the publication of number of research papers in recognized international journals and conferences. His email address is jfss@uninova.pt.

Carlos Agostinho is senior Researcher at UNINOVA-Center of Technology and Systems (CTS) and holds a PhD in the area of industrial information systems by the Faculty of Science and Technology of the NOVA University of Lisbon (FCT/UNL). His main areas of research are Product and Process Meta-modeling, Interoperability and Knowledge Management. He has been actively involved in R&D activities since 2001 and is skillful in the areas of standardization, model-based and semantic interoperability applying MDA/MDI technologies. Currently his research interest is on research in targeting dynamic environments and how principles from complex systems can contribute towards a self-sustainable interoperability. His email address is ca@uninova.pt.

Ricardo Jardim-Goncalves holds a PhD degree in Industrial Information Systems by the New University of Lisbon and is auxiliar professor at the New University of Lisbon, Faculty of Sciences and Technology. He is the coordinator of the research group GRIS (Group for Research on Interoperability of Systems) UNINOVA-Center for Technology and Systems (CTS). His research activities are focused on Standard-based Intelligent Integration Frameworks for Interoperability in Complex Systems, covering System's adaptability in evolutionary and networked environments, architectures, methodologies and toolkits to support improved development, harmonization and implementation of standards for data exchange in industry. He has been technical international project leader for more than 15 years, with more than 80 papers published in conferences, journals and books. He has edited 4 books, and is project leader in ISO TC184/SC4. His email address is rg@uninova.pt.