Towards a Generic Enterprise Systems Architecture Based on Cyber-Physical Systems Principles

Aurelian Mihai Stanescu, Dragos Repta, Mihnea Alexandru Moisescu, Ioan Stefan Sacala, and Monika Benea

University Politehnica of Bucharest, Faculty of Automatic Control and Computers, Bucharest, Romania
{repta_dragos,sacalaioan}@yahoo.com, ams@cpru.pub.ro

Abstract. Systems that can tightly integrate physical with virtual components have represented a priority for the research in the area of ICT. Research efforts have been concentrated in domains such as: Internet of Things, Internet of Services and recently in the domain of Cyber Physical Systems. An important driver for the area of research is represented by the large-scale integration of the physical and cyber worlds. The authors propose a Generic Information System Architecture for Cyber Intelligent Enterprise by taking into account the paradigms of Cyber-Physical Systems

Keywords: Cyber Physical Systems, Cyber Intelligent Enterprise, Future Internet.

1 Introduction

In order to benefit from novel business opportunities and increase their agility, enterprises should adopt a new generation of IT systems capable of (tight interaction) with the physical environment. These context-aware systems come in part as a realization of the pervasive and ubiquitous computing visions. In order to fulfill these novel computing paradigms, the next generation of enterprise information processing systems will have to be able to scale(execute on a spatially distributed array of computing nodes / devices) and to seamlessly integrate a multitude of heterogeneous data streams generated by a large number of connected devices. [9]

Lately, the research effort undertaken toward the achievement of this objective has received a major boost from the research into Cyber-Physical systems and Internet of Things. These two emerging areas of research have slightly different perspectives:

- The Internet of Things (referred as IoT) research is primarily aimed at developing technologies and system architectures, similar to those on which large-scale, loosely coupled systems are being built upon, such as the today's Internet, in order to accommodate uniquely addressable virtual representations of physical devices. [10]
- Cyber-Physical systems on the other hand are focused more on the development of control theory – the study of systems with complex interactions between components unfolding over a wide range of temporal

L.M. Camarinha-Matos and H. Afsarmanesh (Eds.): PRO-VE 2014, IFIP AICT 434, pp. 245–252, 2014.

> and spatial scales. Cyber Physical Systems (referred as CPS) are defined by the National Science Foundation as engineered systems that are built and depend upon the synergy of computational and physical components and unlike the IoT vision, research into CPS doesn't explicitly aim at developing Internet scale systems. [8]

These research directions have covered the problem of system integration from slightly different perspectives, but their ultimate goal can be considered similar taking into consideration the proposed solutions.

In both directions, the research has been mainly driven by the possible future applications that a large-scale integration of the physical and cyber world will enable. Various overview papers in both Internet of Things [4][5] and Cyber Physical Systems [2][3] fields highlighted the benefits and impact that the developed systems might have.

2 Related Work

Research in the area of Future Internet with the main components: Internet of Things (IoT) and Internet of Services has been a priority for FP7 Research Program and is continuing and now integrating with research in the area of Cyber-Physical Systems in the emerging Horizon 2020 Research Programs.

The motivation of developing a Generic Enterprise System Architecture for Future Enterprise [8] is related to the following aspects:

- Enterprise Systems need the means to automatically acquire raw, partially processed or structured data through the help of sensing devices in order to generate information related to the real world.
- The dynamic character of the current economy is determining major changes in the enterprise strategy and organization, triggering new business methods, models and practices which requires agility from Enterprise System.
- Enterprise Systems need to process a rapidly increasing number of new heterogenic parameters and indicators.

Cyber-Physical Systems (CPS) concept integrates:

- Cyber, which refers to computation, communication, and control that is discrete, logical and switched,
- Physical which relates to natural and human-made systems governed by the laws of physics and operating in continuous time.

A generic CPS is composed of: physical objects, sensors, actuators, computing devices (e.g. controllers) and communication networks. [8], [9]

A paradigm shift form Information Society towards Knowledge Society is triggering a technology shift from Intelligent Systems based Enterprise towards Future Internet based Enterprise and Cyber – Physical Systems based Enterprise.

A generic model of the next generation of Enterprise Systems will be introduced, based on the paradigm of Intelligent Cyber-Enterprise. [9]

Intelligent Cyber-Enterprise combines the principles of Cyber Physical Systems and Intelligent Systems in order to facilitate human interaction with both physical and virtual environment, data acquisition and information processing, semantic interoperability and enterprise adaptability.

3 Generic System Architecture for Cyber – Intelligent Enterprise

In order to develop a capable (/viable/feasible) Cyber Intelligent System [8], [9], we consider the following major aspects:

- Physical device abstraction
- Information processing
- Domain representation

The system will need to employ powerful abstractions in order to be able to create accurate virtual representations of the devices that interact with the environment – sensors and actuators and their capabilities. Depending on the system's objectives, various abstractions can be used for these devices, but mainly the service computing paradigm is employed.

Domain representation is an important aspect of any system that integrates physical and virtual resources. It is essential for dealing with the heterogeneity of the system's components and the complexity of the environment. Following the large-scale adoption of Semantic Web technologies, the system's domain is usually represented through a set of ontologies – both generic and system specific. An under-explored aspect of domain representation is its dynamic nature. The domain representations evolve with the system as it gains concepts which increase its expressive power, allowing the system to deal with new situations. The evolution of the system's domain can be enabled by the information processing capabilities of the system such as determining new relationships between concepts, creation of new concepts and rules, importing schemas from legacy data processing systems such as relational or XML databases.

Taking into account these three functional aspects, we propose a generic architecture for a Cyber Intelligent Enterprise [9] system. The proposed architecture will have a layered structure:

- The lower level of the system contains the system's interface with the physical world / physical phenomena represented by the enabled devices – sensors and actuators – and the communication infrastructure required to connect them to the upper level
- The intermediate middleware layer that implements common operation related to the roles of a Cyber-Physical System
- The upper layer contains the systems applications that implement the domain specific logic(system specific objectives)

A generic architecture of a Cyber Intelligent Enterprise is presented in Fig1.

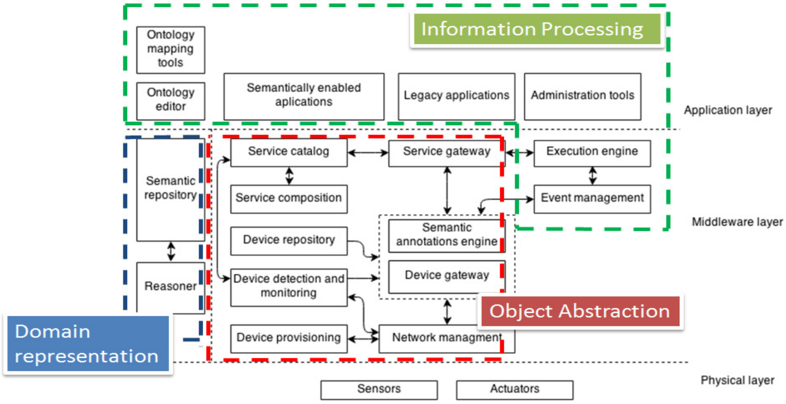

Fig. 1. The three main roles of each IoT / Cyber-Physical system

The middleware layer contains the following components:

- Network management components that manage all low level aspects of the communication with the enabled devices, such as network formation, device discovery and data transfer

- Device detection and monitoring component facilitates the implementation of self-configuration and adaptive behaviors. Using existing protocols such as Universal Plug and Play (UPnP), WS-Discovery, the component will detect when devices join or leave the network and publish the corresponding events to relevant system components or applications thus enabling the deployment of fallback strategies.

- The device gateway is an important component for realizing the physical device abstraction aspect of the system as it manages the instances of the device wrapping components. Based on device joining events generated by the device detection and monitoring" component, the device gateway will fetch and instantiate the corresponding device wrapping components from the device repository. These instances will be removed when the system detects that the devices are no longer present in the managed network. The device gateway contains a rule-based semantic annotator that updates the messages relayed to the upstream components with annotations referring to the system's ontology

- The device provisioning component allows the system to alter the pre-defined capabilities of the physical devices through firmware, a method already proved feasible [7][8]

- The device repository component manages the library to wrapper component for the devices supported by the system. Each of these specialized components must hide the device's proprietary interfaces and expose instead a uniform, semantically-enabled interface.

- Service gateway – manages the endpoints of the services defined in the system and represents the main access point for the top level applications to the available resources. In order to lower the requirements for the top level applications / to ease the process of integrating existing applications into the proposed system, the service gateway component should be able to handle various types of semantically enhanced web services such as OWL-S, WSMO, or semantically enabled REST – SA-REST

- Service composition – allows the creation of new services based on the existing resources / services available in the system with the aim of fulfilling a user specified objective. In order to simplify the underlying planning task, a user defined abstract composition template may be used. Early research into the problem of composing physical services, such as [6], has delivered encouraging results.

- Service catalog – represents the central repository of services for a node of the system. Novel service query and selection methods are required in order to deal with services referring to physical objects. [7].

- Event management – facilitates the usage of asynchronous, event-driven interactions between the system's components (and/or applications) by implementing the role of an event broker. Thus, it will accept events published by system components and route them to interested components that have previously registered matching subscriptions.

- Rule based execution engine – executes and manages a repository of simple, event-driven programs. It simplifies the deployment of new system behaviors by removing the execution responsibility from other components. These event-driven programs will be able to consume event, publish new ones and call any service registered in the system's service catalog. For describing these simple programs, a simple and portable language can be used, such as the SCXML (State Machine Notation for Control Abstraction), a new W3C submission.

- Semantic repository – stores and manages the system's ontology that includes not only the concepts required to describe the physical and virtual processes with which the system interacts, but also instances that represent system objects such as services, enabled devices, system components, etc. The semantic repository uses an external semantic reasoner for the deduction of implicit relationships between ontological concepts and to insure the validity of the representation.

In order for an Enterprise to be able to adapt to the changing environment, data collected from sensors must generate events leading to changes to the Enterprise Business Process.

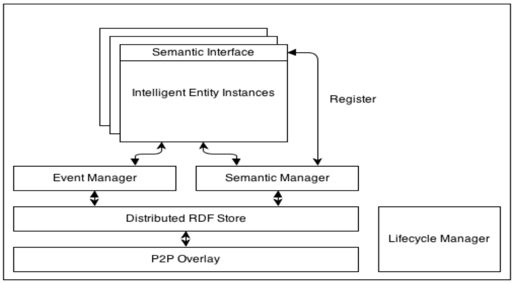

Fig. 2. Semantic interfaces representation [9]

With the vertical evolution of control system, up to the concept of Cyber-Physical Systems, the systems gained more autonomy by modeling more complex behaviors.

4 Enterprise System Based on Generic System Architecture

In this section the authors propose a fully-distributed ontology based implementation based on the Generic System Architecture presented in the previous section.

An important aspect of the system is represented by the ontology. The system's ontology is used to store not only the concept that describe the processes, virtual or physical, that the system interacts which, but also its structure, state and the available interaction mechanisms. As such, several classes in the ontology will be used to describe the structural aspects of the system such as Device, Component, Application and the interactions between components: Service, EventSink, EventSource. This approach will enable rapid development of flexible systems.

A distributed deployment of the system will require that a distributed RDF store is used as "Semantic Repository", allowing the storage of RDF triples in a structured P2P overlay and the execution of SPARQL queries with RDFS reasoning. A Distributed RDF Store component is created for each node.

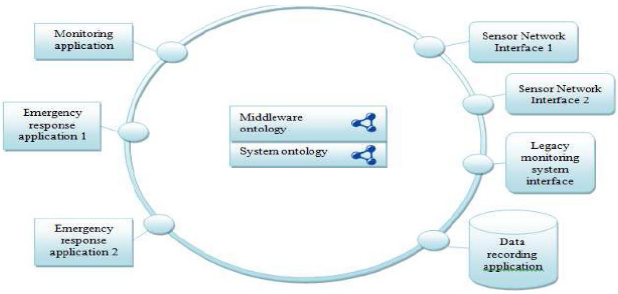

Fig. 3. Node block diagram

This component can also be used to build an implementation of the "Event Subscription Management" and "Event Routing" components, by applying the Triplespace computing paradigm. This will require the implementation of a continuous query processing algorithm, applications can register SPARQL queries and be notified when the selected information changes. The subscriptions will be queries selecting "EventSource" semantic instances. The Event Manager will transmit the event generation rule from event sinks to the event sources and will ensure the correct registration in the Distributed RDF Store of the semantic instances representing current event sources, event sinks and events. The Intelligent Entity Repository is a separate component in this implementation.

The "Semantic Manager" component: provides an interface to the Distributed RDF Store with OWL reasoning, allows the registration of the elements of the Semantic Interface exposed by each Intelligent Entity.

As mentioned in the previous, section, the system will allow the rapid deployment of simple, event-driven programs that will be stored and executed in the "Execution Engine" component. A snippet form one such program written in an extended version of the SCXML notation is described in "Listing 1". The program registers and event sink (event subscription) for the events generated by temperature sensors from the location with code "123" and an event source referencing the same location. The event source defines the channel through which the deployed behavior will generate its own events. The “Behavior Execution Engine” analyzes the content of the SCXML document on deployment and adds the semantic definitions contained by the “sinks” and “sources” elements in the system’s ontology. The events will be routed to the “Behavior Execution Engine” with the assistance of the “Event Manager” that processes the event sink definitions and transforms them into event subscriptions.

```
<state id="idle">
            <transition event="EventSink1"
cond="#{event.hasEventData.hasTemperature &gt; 25}"
target="alert" />
          </state>
          <state id="alert">
            <onentry>
              <send event="VirtualSensorSource1">
                <sys:EventData>
                  <sys:hasAlert>
                    <sys:Alert>
                      <sys:alertLevel>WARNING</sys:alertLevel>
<sys:message>Elevated temperature at location
&quot;123&quot;</sys:message>
                    </sys:Alert>
                  </sys:hasAlert>
                </sys:EventData>
              </send>
            </onentry>
                     …..
          </state>
```

Fig. 4. Listing 1

In the current example, the temperature events routed to the behavior’s event sink can be used to transition from an “idle” to an “alert” state, as shown in Listing 1. The transition will happen when an event received through the previously defined “EventSink1” validates the condition “temperature above 25”. When the state of the defined behavior becomes “alert”, the “Behavior Execution Engine” will generate an alert event associated with event source “VirtualSensorSource1”. The event will convey the information contained by the “EventData” OWL Individual defined inside the “send” element.

5 Conclusions

A Generic System Architecture for Cyber Intelligent Enterprise must include and not restrict to capabilities such as: semantic concepts for the description of system components, support for both asynchronous and synchronous communication and a distributed storage of the system’s semantic concepts.

An application based on Generic System Architecture for Cyber Intelligent Enterprise has been implemented and tested using a predefined scenario. Comparing the system's behavior to similar systems based on currently available technologies, we proved the feasibility of the middleware's architecture and its advantages. Further research related to system performance is being conducted in order to prepare for "real-life" enterprise conditions.

Acknowledgement. The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132397.

References

1. Afsarmanesh, H., Camarinha-Matos, L.M.: Management of Information Supporting Collaborative Networks. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 1–6. Springer, Heidelberg (2009)
2. Barnaghi, P., Wang, W., Henson, C., Taylor, K.: Semantics for the Internet of Things: early progress and back to the future. International Journal on Semantic Web and Information Systems (IJSWIS) 8(1), 1–21 (2012)
3. Rajkumar, R.R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th Design Automation Conference, pp. 731–736. ACM (2010)
4. Bandyopadhyay, D., Sen, J.: Internet of things: Applications and challenges in technology and standardization. Wireless Personal Communications 58(1), 49–69 (2011)
5. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks 54(15), 2787–2805 (2010)
6. Meyer, S., Sperner, K., Magerkurth, C., Pasquier, J.: Towards modeling real-world aware business processes. In: Proceedings of the Second International Workshop on Web of Things, p. 8. ACM (2011)
7. Kostelnik, P., Sarnovsk, M., Furdik, K.: The semantic middleware for networked embedded systems applied in the Internet of Things and Services domain. Scalable Computing: Practice and Experience 12(3) (2011)
8. Dumitrache, I., Caramihai, S.I., Stanescu, A.: From Mass Production to Intelligent Cyber-Enterprise. In: 2013 19th International Conference on Control Systems and Computer Science (CSCS), pp. 399–404 (2013) Print ISBN: 978-1-4673-6140-8
9. Sacala, I S., Moisescu, M.A., Repta, D.: Towards the Development of the Future Internet Based Enterprise in the Context of Cyber-Physical Systems. In: 2013 19th International Conference on Control Systems and Computer Science (CSCS), pp. 405–412. IEEE
10. Moisescu, M.A., Sacala, I.S., Stanescu, A.M., Serbanescu, C.: Towards Integration of Knowledge Extraction Form Process Interoperability in Future Internet Enterprise Systems. Information Control Problems in Manufacturing 14(1), 1458–1463 (2012)
11. Stegaru, G., Danila, C., Sacala, I.S., Moisescu, M.A., Stanescu, A.M.: E-Services quality assessment framework for collaborative networks. Journal of Enterprise Information Systems, Taylor and Francis (2014), doi:10.1080/17517575.2013.879213