Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Towards the development of a Cyber-Intelligent Enterprise System Architecture

Dragos Repta*, Ioan Stefan Sacala*, Mihnea Alexandru Moisescu*, Aurelian Mihai Stanescu*

** Faculty of Automatic Control and Computers Science, University Politehnica of Bucharest, 313 Splaiul Independentei, Sector 6,Bucharest, Romania*

(e-mail: repta_dragos@yahoo.com, sacalaioan@yahoo.com, mamihnea@gmail.com, ams@cpru.pub.ro)

Abstract: Although the concept of Cyber Intelligent Enteprise is largely based on Cyber Physical Systems (CPS), research in adjacent fields, like Internet of Things (IoT), is also important as it provides a valuable set of system architectures and integration techniques.

The current socio-economic context provides the perfect opportunity for the development of a new breed of enterprise system architecture – Cyber Intelligent Enterprise.

In order to design a generic architecture for Cyber Intelligent Enterprise system we introduce the concept of Intelligent Entity. Based on the generic architecture presented in the previous section, we propose an implementation for the "Integration Layer".

Keywords: Cyber Physical Systems, Cyber Intelligent Enterprise, Enterprise System Architecture

1. INTRODUCTION

Cyber-Physical Systems (CPS) represents a newly developed concept of systems with tightly integrated computational, physical and communication processes (http://cyberphysicalsystems.org/).

While many current systems contain software, networking and physical components, they are designed and integrated in an independent manner, without using a global system model. In contrast with this approach, the CPS vision aims to develop a unified framework for modelling and implementing complex systems as a whole.

Although the research field of Cyber-Physical Systems has been only recently established, several surveys (Baheti et al 2001, Sha et al 2009, Shi et al 2011 and Wan et al 2011) argue that the unified approach to system design will have a significant positive impact in various areas such as healthcare, energy distribution, transport system, etc.

In this context, the concept of Cyber Intelligent Enterprise (Dumitrache 2013) was developed, as the direct application of the CPS vision in the field of enterprise systems. Tight integration of physical devices and business processes will enable new capabilities and increase the enterprise agility (Overby 2006). We propose a generic architecture for a Cyber Intelligent Enterprise system, based on a set of technologies developed in the fields of Internet of Things and Semantic Web.

2. STATE OF THE ART

Although the concept of Cyber Intelligent Enterprise is largely based on CPS, research in adjunct fields, like Internet of Things (IoT), is also important as it provides a valuable set of system architectures and integration techniques.

Various IoT research projects have developed middleware architectures able to encapsulate physical devices – sensors, including RFID readers, and actuators – and expose them as virtual resources that can be used by IT systems (Katasonov et al 2008, Kostelnik et al 2011, De Souza et al 2008).

Another important driver for the development of Cyber Intelligent Enterprise systems has been the research into semantic technologies that led to the creation of the Semantic Web vision (Berners-Lee 2001). Among these, the most relevant to our research has been the creation of standard formats for ontologies, machine-interpretable representations of domain knowledge. The appearance of standard formats for representing ontologies has significantly simplified their creation and integration into IT systems. As ontologies are fundamental for the achievement of semantic interoperability (Tolk 2003), systems using the semantic concepts encapsulated by them can easily integrate heterogeneous components and data sources. (Moisescu, 2012)

Finally, the development of enterprise CPS solutions – Cyber Intelligent Enterprises – will also benefit from the research into WSN (Wireless Sensor Networks) and MANET (Mobile Ad-hoc Networks) (Wu 2011). In recent years, the development of low-powered electronic devices, batteries and efficient wireless communication protocols, has led to the creation of viable WSN architectures that can be used to implement self-organizing, flexible, inexpensive and robust wireless data gathering networks. (Sacala, 2013)

In summary, the development of the CPS concept and, implicitly, the Cyber Intelligent Enterprises, has been facilitated by a series of enablers:

WSN / MANET:

- Low power microprocessors / microcontrollers and improved battery technology

- Efficient communication and routing protocols designed for low-powered devices and ad-hoc networks that allow self-organized and flexible networks

- Algorithms for the dissemination of collected data in the network

Internet of Thing - Reference architectures for integrating physical devices in existing IT systems

Semantic Web – Semantic technologies (ontologies, semantic annotation standards for data and business objects, semantic query languages) that allow interoperability at semantic level, thus alleviating the data heterogeneity issue.

The generic architecture proposed in this paper incorporates some of the previously mentioned enabling technologies and follows the research into a new generation of enterprise systems (Dumitrache 2008) that lead to the concept of Cyber Intelligent Enterprise (Dumitrache 2013).

3. GENERIC ARCHITECTURE FOR CYBER INTELLIGENT ENTERPRISE SYSTEMS

The current socio-economic context provides the perfect opportunity for the development of a new breed of enterprise system architecture – Cyber Intelligent Enterprise:

- The creation of new business opportunities
- Increasing enterprise agility (Overby 2006) – the implementation of Cyber Intelligent Enterprise concept will increase both the sensing component and the response component of enterprises

The main characteristics of Cyber Intelligent Enterprise systems are:

- Capable of using both virtual and physical resources
- Scalable
- Capable of processing vast amounts of heterogeneous information

In order to design a generic architecture for Cyber Intelligent Enterprise systems we introduce the concept of Intelligent Entity (Fig. 1).

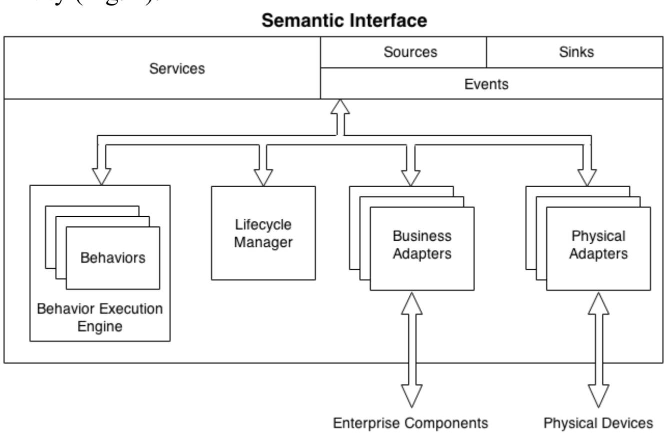

Fig. 1: Generic Cyber Intelligent Enterprise System Architecture

As depicted in the diagram, each Intelligent Entity will contain the following components:

Semantic interface – provides a unique way of accessing both virtual and physical resources of the system. This interface will allow both service and event based interactions between the system's components because, as some authors argue (Kong 2009, Wieland 2009), in today's complex business environments, they are complementary. Each element of this interface – services, event sinks and event sources – will have a semantic description attached and all information passed through it is semantically enhanced.

Business adapters – Each Intelligent Entity can host a series of specialized components that will connect to existing enterprise components like SOA enabled applications, business process execution engines, service orchestration engines, etc. and expose their operation through the unique semantic interface. These business adapters will need to be built for each enterprise component that needs to be integrated in the system, as each will require a different connection mechanism and data transformation rules for the semantic enrichment of data.

Physical adapters – the Intelligent Entities can also host a set of adapters for the physical devices. Each Physical Adapter will need to handle all aspects of device discovery, configuration, communication and data transformation.

Behaviour Execution Engine – instances of intelligent Entities will be able to execute behaviors expressed in various languages. For each supported language, the Intelligent Entity instance will have a corresponding execution engine. For input and output, the stored behaviours will use the semantic interface.

Lifecycle Manager – handles all aspects of the Intelligent Entity's lifecycle - initializing, clean-up, monitoring the state of the components hosted on the current instance and deploying new behaviours or adapters. The lifecycle manager can expose a set of operations through the semantic interface, such as a services to deploy / un-deploy components running on the current instance, or an event source through which it will generate events regarding the change in the state of components (for example, generating an event when a business adapter stops due to a connection error to the enterprise component). The UML State diagram in Fig. 2 depicts a possible implementation of the Lifecycle Manager.

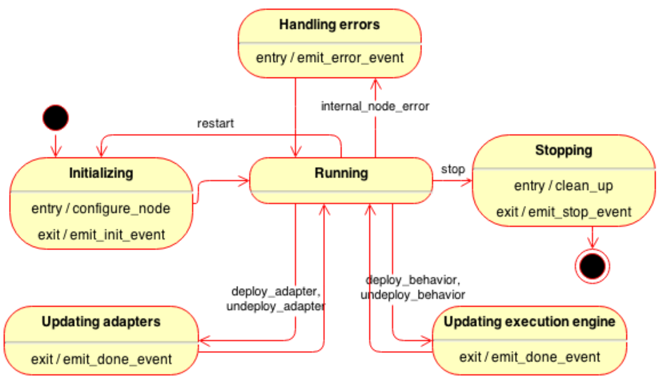

Fig. 2: Possible implementation of Lifecycle Manager

Based on the proposed concept of Intelligent Entity, we developed a generic architecture of a Cyber Intelligent System (Fig. 3).

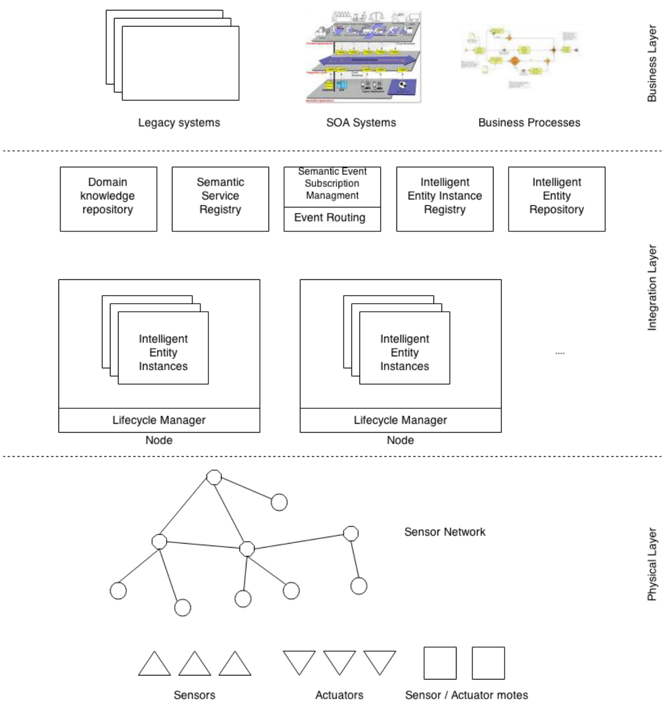

Fig. 3: Generic Cyber Intelligent System Architecture

As presented in the above diagram, a Cyber Intelligent System has 3 layers:

Business layer - contains all the elements that are currently used to develop enterprise systems: SOA components, business process execution engines, domain specific business applications (ERP, etc.)

Physical Layer - contains all the system's components that collect data from the physical environment (sensors), interact with the environment (actuators) and relay the data and commands to and from the upper level (wired or wireless networking components).

Integration Layer - has the role of integrating data from the upper and lower levels of the system and thus, to provide the foundation for the development of Cyber Intelligent Systems. The layer incorporates a homogeneous structure of computational nodes hosting Intelligent Entity Instances and a set of support components needed to manage this structure.

In order to implement the Integration Layer the system needs to contain a set of support components that have to fulfil the following roles:

Nodes – computational units that host the Intelligent Entity Instances. All the aspects of the node's lifecycle such as initialization (including initial configuration), clean-up, monitoring, etc. are managed by the "Lifecycle Manager". Each node will expose its own Semantic Interface appearing in the system as an Intelligent Entity. Through the exposed interface, the node can receive command such as start / stop / deploy new instance or generate events regarding its state.

Domain knowledge repository – handles a machine-interpretable abstraction of the system's domain. The elements of the domain must be uniquely addressable so that the added metadata (by the physical and business adapters) can properly refer to the domain's concepts. Furthermore the relations between the concepts in the system's domain will have to be explicitly stated and the "Domain knowledge repository" will need to provide reasoning capabilities so that the information consumers in the system will be able to accurately determine the context in which the information needs to be processed. The system's domain can be expressed using any formalism that respects the mentioned conditions: taxonomies, ontologies, etc.

Semantic service repository – stores the semantic description of all the services exposed by the Intelligent Entities instances. Each service description will be enhanced with metadata referring to the system's knowledge base. The semantically enhanced services can be described using existing standards such as OWL-S, SAWSDL, etc.

Semantic Event Subscription Manager / Event routing – handles the subscriptions of the system's components to the semantic events generated by the Intelligent Entities. The subscriptions will select events based on their context and will specify an event generation rule – for example: subscribe to events generated by the temperature sensors from area X with the event generation rule "onValueChange". The information regarding the subscriptions to the events will be transmitted to the "Semantic Event Router" which will ensure the transfer of information between the event generator and the subscriber.

Intelligent Entity Instance Registry – stores all the information (state, location, current configuration) regarding the currently deployed Intelligent Entities Instances and provides a shortcut to the lifecycle management services exposed by the Intelligent Entities.

Intelligent Entity Repository – handles the process of creation and deployment / un-deployment of Intelligent Entities. Due to their modular structure, the Intelligent Entities will be able to be created at run-time using existing libraries of Business and Physical Adapters. Furthermore, this component will also manage the deployment of (new) behaviors on existing Intelligent Entities.

4. CYBER INTELLIGENT ENTERPRISE SYSTEM REALIZATION

Based on the generic architecture presented in the previous section, we propose an implementation for the "Integration Layer".

In the proposed implementation, the system's ontology will play a central role as it will be used to store the system's structure and will support the transfer of information between the system's components.

In order to provide a consistent representation of the system, the ontology will be based on a set of preexisting semantic definitions that form the system's built-in ontology (fig. 4).

Based on these semantic definitions, the system will maintain a semantic representation of its components (Intelligent Entities, Nodes, and Adapters) and their interactions (Services, Event Sinks, Event Sources and Events).

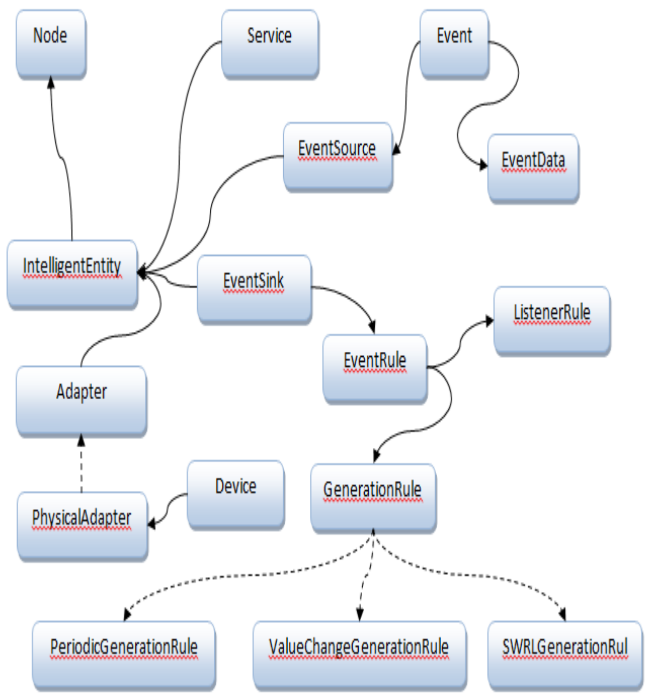

Fig. 4: Built-in ontology for Cyber Intelligent Enterprise

Using this strategy, there is no need for specialized registry components, as information about the current Intelligent Entity instances, services and event sources can be obtained by executing SPARQL queries on the system ontology.

For the role of the "Domain Knowledge Repository", a distributed RDF store like the Atlas system (Kaoudi 2010) can be used. This system allows the storage of RDF triples in a structured P2P overlay and the execution of SPARQL queries with RDFS reasoning.

Structured P2P overlays based on distributed hash tables (DHT), like Bamboo DHT, have remarkable characteristics that allow the development of efficient, robust and scalable distributed systems (Aberer 2005). The fact that the Atlas system leverages these properties by being designed to use such an overlay was one of the main reasons why it was chosen.

Each node in the integration layer will have its own Distributed RDF Store component (that nodes will form the P2P Overlay).

The distributed RDF Store can also be used to fulfill the roles of the "Event Subscription Management" and "Event Routing" through the application of the Triplespace computing paradigm (Fensel 2004). This concept is a semantic interpretation, based on the RDF triple concept, of the tuplespace-based computing paradigm. In this paradigm, the communication between the components of a system is done by reading and writing in a shared space instead of passing messages.

In order for the distributed RDF store to fulfil this role, it requires a continuous query processing algorithm (Liarou 2007). Using this algorithm, the system's components can register SPARQL queries and be notified when the selected information changes.

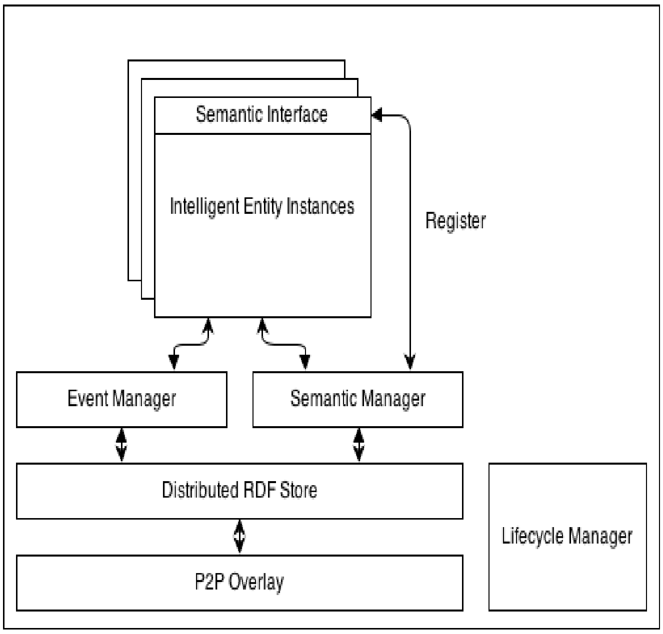

Fig. 5: Block diagram representation of a node for an Intelligent Entity

As we mentioned in the previous section, the event subscriptions refer to the context in which the event is generated – in this case the subscriptions will be queries selecting "EventSource" semantic instances.

For the system to be complete there will be an additional component – Event Manager – that 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.

Of the 5 support components of the Integration Layer, only the Intelligent Entity Repository will remain a separate component in this implementation. The roles of the rest will be taken by the Distributed RDF Store, Event Manager and Semantic Manager components. The „Semantic Manager" component will provide an interface to the Distributed RDF Store with OWL reasoning and allow the registration of the elements of the Semantic Interface exposed by each Intelligent Entity. All these components will be deployed at each node in the Integration Layer. A block diagram of a node is presented in Fig. 5.

The „Behaviour Execution Engine" of each Intelligent Entity will support several methods of defining behaviours. For example, behaviours can be specified as state machines using an extended version of the SCXML notation (http://www.w3.org/TR/scxml/). The defined state machines will be driven by the events that are generated in the system.

```
<scext:sources>
        <owl:NamedIndividual
rdf:about="&sys;VirtualSensorSource1">
                <rdf:type
rdf:resource="&sys;EventSource"/>
```

```
        <sys:hasLocation><sys:Location><sys:hasLegacyCode>123</sys:hasLegacyCode></sys:Location></sys:hasLocation>
        </owl:NamedIndividual>
</scext:sources>

<scext:sinks>
        <owl:NamedIndividual
rdf:about="&sys;EventSink1">
                <rdf:type rdf:resource="&sys;EventSink"/>
                <sys:hasEventRule>
                        <sys:EventRule>
                                <sys:hasGenerationRule>

        <sys:PeriodicGenerationRule>

        <sys:interval
rdf:datatype="&xsd;integer">20</sys:interval>

        </sys:PeriodicGenerationRule>

        </sys:hasGenerationRule>
                                <sys:hasListenerRule>

        <sys:ListenerRule>

        <sys:eventSourceQuery>SELECT ?source WHERE
{
                                                        ?source
rdf:type sys:EventSource ;

          sys:hasMeasurement &quot;temperature&quot; ;

          sys:hasLocation ?loc .
                                                        ?loc
sys:hasLegacyCode &quot;123&quot; .

        }</sys:eventSourceQuery>

        </sys:ListenerRule>
                                </sys:hasListenerRule>
                        </sys:EventRule>
                </sys:hasEventRule>
        </owl:NamedIndividual>
</scext:sinks>
```

Listing 1

The extensions to the SCXML notation will be used to define the „Semantic Interface" elements associated with the defined behaviors. In „Listing 1", we present a snippet from a SCXML document that defines an event sink for events generated by temperature sensors from location „123" and an event source for the same location. The event source defines the channel through which the deployed behavior will generate its own events. The „Behavior Execution Engine" will analyze the content of the SCXML document on deployment and add the semantic definitions contained by the „sinks" and „sources" elements in the system's ontology. Furthermore, the event sink definitions will be processed by the „Event Manager" and transformed into event subscriptions so that the required events will be routed to the „Behavior Execution Engine".

```
        <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>
```

Listing 2

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 2. 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 (in this case, an alert that the temperature in location "123" is above 25).

5. CONCLUSIONS AND FUTURE WORK

Based on the recent developments in various research fields such as Cyber Physical Systems, Internet of Things and Semantic Web, in this article, we propose a new system architecture that enables the tight integration of physical components with business processes in enterprise systems. The adoption of this type of system can have a significant positive impact as it creates new business opportunities and increases the agility of companies. Aiming for a flexible and scalable architecture, we introduce the concept of Intelligent Entity as the primary building block of Cyber Intelligent Enterprise systems. The primary roles of

this software component will be to facilitate the access to the system's resources and to enable rapid deployment of new behaviours.
Based on the developed generic architecture, in the last section of the paper, we present an implementation example in which the system's components and their interactions are maintained in a system-wide ontology.

We aim to further research this concept by fully developing all the required software components and validating the concept in real-world situations.

REFERENCES

Baheti, R., & Gill, H. (2011). Cyber-physical systems. *The Impact of Control Technology*, 161-166.

Sha, L., Gopalakrishnan, S., Liu, X., & Wang, Q. (2009). Cyber-physical systems: A new frontier. *In Machine Learning in Cyber Trust* (pp. 3-13). Springer US.

Shi, J., Wan, J., Yan, H., & Suo, H. (2011, November). A survey of cyber-physical systems. *In Wireless Communications and Signal Processing (WCSP)*, 2011 International Conference on (pp. 1-6). IEEE.

Wan, J., Yan, H., Suo, H., & Li, F. (2011). Advances in Cyber-Physical Systems Research. *TIIS*, 5(11), 1891-1908.

Overby, E., Bharadwaj, A., & Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, 15(2), 120-131.

Katasonov, A., Kaykova, O., Khriyenko, O., Nikitin, S., & Terziyan, V. Y. (2008). Smart Semantic Middleware for the Internet of Things. *ICINCO-ICSO*, 8, 169-178.

Kostelnik, P., Sarnovsk, M., & Furdik, K. (2011). The Semantic Middleware for Networked Embedded Systems Applied in the Internet of Things and Services Domain. Scalable Computing: Practice and Experience, 12(3).

De Souza, L. M. S., Spiess, P., Guinard, D., Köhler, M., Karnouskos, S., & Savio, D. (2008). Socrades: A web service based shop floor integration infrastructure. *In The internet of things* (pp. 50-67). Springer Berlin Heidelberg.

Wieland, M., Martin, D., Kopp, O., & Leymann, F. (2009, January). SOEDA: A method for specification and implementation of applications on a service-oriented event-driven architecture. *In Business Information Systems* (pp. 193-204). Springer Berlin Heidelberg.

Kong, J., Jung, J. Y., & Park, J. (2009). Event-driven service coordination for business process integration in ubiquitous enterprises. Computers & Industrial Engineering, 57(1), 14-26.

Kaoudi, Z., Koubarakis, M., Kyzirakos, K., Miliaraki, I., Magiridou, M., & Papadakis-Pesaresi, A. (2010). Atlas: Storing, updating and querying RDF (S) data on top of DHTs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4), 271-277.

Aberer, K., Alima, L. O., Ghodsi, A., Girdzijauskas, S., Haridi, S., & Hauswirth, M. (2005, August). The essence of P2P: A reference architecture for overlay networks. *In Peer-to-Peer Computing*, 2005. P2P 2005. Fifth IEEE International Conference on (pp. 11-20). IEEE.

Fensel, D. (2004). Triple-space computing: Semantic Web Services based on persistent publication of information. *Intelligence in Communication Systems*, 43-53.

Liarou, E., Idreos, S., & Koubarakis, M. (2007). Continuous RDF query processing over DHTs. *In The Semantic Web* (pp. 324-339). Springer Berlin Heidelberg.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, 284(5), 28-37.

Tolk, A., & Muguira, J. A. (2003, September). The levels of conceptual interoperability model. *In Proceedings of the 2003 Fall Simulation Interoperability Workshop* (Vol. 7).

Wu, F. J., Kao, Y. F., & Tseng, Y. C. (2011). From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4), 397-413.

Dumitrache, I., Caramihai, S. I., & Stanescu, A. (2013). From Mass Production to Intelligent Cyber-Enterprise. In Control Systems and Computer Science (CSCS), 2013 19th International Conference on (pp. 399-404). IEEE.

Dumitrache, I., & Caramihai, S. I. (2008). Towards a new generation of intelligent manufacturing systems. In Intelligent Systems, 2008. IS'08. 4th International IEEE Conference (Vol. 1, pp. 4-33). IEEE.

Dumitrache, I; Constantin, N. (2013). Some Challenges for the Cyber Physical Energy System; in Convergence of Information Technologies and Control Methods with Power Systems, Volume # 2 | Part# 1, pg 1-6; Digital Object Identifier (DOI) 10.3182/20130522-3-RO-4035.00045

Stegaru G., Danila C., Sacala IS, Moisescu M., Stanescu AM. (2014) E-Services quality assessment framework for collaborative networks, Enterprise Information Systems, Taylor & Francis 2014, DOI: 10.1080/17517575.2013.879213

Sacala IS, Repta D., Moisescu MA, (2013) Towards the Development of the Future Internet Based Enterprise in the Context of Cyber-Physical Systems, Control Systems and Computer Science (CSCS), 2013, ISBN: 978-1-4673-6140-8; DOI: 10.1109/CSCS.2013.84

Moisescu MA, Sacala IS, Stanescu AM, Serbanescu C., (2012) Towards Integration of Knowledge Extraction Form Process Interoperability in Future Internet Enterprise Systems, Information Control Problems in Manufacturing, Volume # 14 | Part# 1, Page Numbers: 1458-1463 DOI 10.3182/20120523-3-RO-2023.00395