# **OSMOSE: A Paradigm for the Liquid-Sensing Enterprise**

Sergio Gusmeroli TXT e-solutions, 20126 Milan, Italy sergio.gusmeroli@txtgroup.com Carlos Agostinho Catarina Lucena Centre of Technology and Systems, Centre of Technology and Systems, CTS, Uninova, 2829-516 Caparica, CTS, Uninova, 2829-516 Caparica, Portugal Portugal ca@uninova.pt

Michele Sesana TXT e-solutions, 20126 Milan, Italy michele.sesana@txtgroup.com Artur FelicKlaus FischerCAS Software AG, CAS-Weg 1-5, DFKI GmbH, 66123 Saarbrücken,<br/>76131 Karlsruhe, Germany<br/>Artur.Felic@cas.deGermany<br/>Klaus.Fischer@dfki.de

Gabriella Monteleone Piksel, Via Breda 176, Milan, 20126 Italy gabriella.monteleone@piksel.com

# Abstract

With the spreading of sensors networks, the management of business reality has been progressively changed exploiting the inherent features of the future IoT and smart objects. Today's enterprises need to become self-aware not only in terms of their networked ecosystem but also in face of their inner sub-systems and devices. Indeed, the integration of service and event technology, smart agents and "smart objects" enables enterprise innovation and competitiveness in a globalized economy. Those enterprises would be capable of sensing and reacting from physical to complex business incitements, maximizing their innovation potential and sustaining interoperability along the operational life cycle. This paper presents the OSMOSE paradigm for the liquid-sensing enterprise, a novel concept being developed in the context of the Future Internet to enable new business models for the sensing enterprise. Initial developments and a conceptual architecture are presented and illustrated with the help of an on-going industrial scenario.

# **1. Introduction**

According to the FInES research roadmap 2025 [1], *sensing enterprise* and *liquid enterprise* are two qualities of being which are considered strategic for any future enterprise. The sensing enterprise will emerge with the evolution of the Internet of things (IoT), when objects, equipment, and technological infrastructures will exhibit advanced networking and processing capabilities, actively cooperating to form a sort of 'nervous system' within the enterprise

Copyright © 2015 by the paper's authors. Copying permitted only for private and academic purposes.

In: M. Zelm (eds.): Proceedings of the 6<sup>th</sup> Workshop on Enterprise Interoperability, Nîmes, France, 27-05-2015, published at http://ceur-ws.org

next generation [2].

The concept represents a fundamental change in the business models and information systems that is not immediate, and should be supported by methods and tool capable of supporting the evolution of the traditional organizations towards the tremendous possibilities offered by the IoT-enabled worlds. Indeed, latest research in the area is making quite clear that the take-up of the sensing enterprise concept will enable very advanced and promising new business models and applications thanks to the adoption of FI technologies. Research initiatives such as the OSMOSE (www.osmose-project.eu) or Proasense (www.proasense.eu) European projects as well as dedicated scientific sessions and working groups such as the one held in the last IFAC World Congress [3] promoted a wide debate on the need for a convincing unifying holistic model for a digital sensing enterprise as well as a common reference architecture for next generation enterprise applications based on the IoT and other FI technologies and generic enablers.

This paper explores this need and presents the initial developments of a conceptual model and reference architecture for the liquid-sensing enterprise, a specification of the sensing enterprise concepts being developed with the support of project OSMOSE (Osmosis Applications for the Sensing Enterprise). The main objective of the project is to develop a reference architecture, a middleware and some prototypal applications for the *sensing-liquid enterprise*, by interconnecting *real*, *digital*, and *virtual worlds* on the same way a semi-permeable membrane permits the flow of liquid particles through itself. After an initial assessment of the OSMOSE metaphor, a brief state of the arte regarding event driven architectures is made. In section 4, the conceptual architecture for the OSMOSE liquid sensing enterprise is presented and section 5 presents some of the deployed middleware and implementations being conducted. Section 6 concludes and presents some future work.

### 2. The OSMOSE Metaphor

The *liquid enterprise* can be considered as an enterprise having fuzzy boundaries, in terms of human resources, markets, products and processes. Its strategies and operational models will make it difficult to distinguish the 'inside' and the 'outside' of the company[1]. This concept can be better explained if a metaphor from physics is adopted. Let us imagine that the *liquid-sensing enterprise* is, in fact, a pot internally subdivided into three sectors by means of three membranes and forming the real-digital-virtual sector [4][5]. Like is represented in Fig. 1, a blue liquid is poured into the bottom section (real world population), a red liquid to the middle section (digital world population) and a green liquid into the third sector (virtual world population).

If the membrane is totally **impermeable** (left part of Fig. 1), the population of the three liquids (worlds) will never mix together and, if they want to communicate, they need to send blind messages across the membranes. This meats the classical definition of interoperability, which is defined as the ability of disparate and diverse organizations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between them [6]. However, in those interoperability scenarios, the two or more actors are totally independent entities (e.g. two or more enterprises, organizations, or even people and objects).

On the other hand, if the membranes are **semi-permeable**, by following the rules of osmosis, each of the world's population could pass through the membrane and influence the neighbouring world, so that in reality in the blue real world is possible to find a red-green shadow ambassadors of the digital/virtual world and similarly for the other worlds (see right part of Fig. 1).

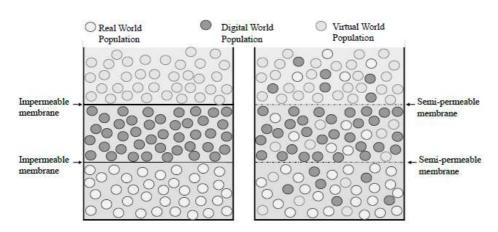


Fig. 1: Liquid-Sensing Enterprise physics metaphor

## 2.1. IoT enabled Manufacturing LSE Framework

The European Research Cluster on the Internet of Things (IERC) recognizes that ICT development is generating more and more things/objects embedded with sensors, which are gaining the ability to communicate with each other. Hence, IoT is enabling objects in our environment to become active systems, sharing information with many stakeholders, and gaining the skills for recognizing events in their surroundings, to which they are acting and reacting autonomously [7][4].

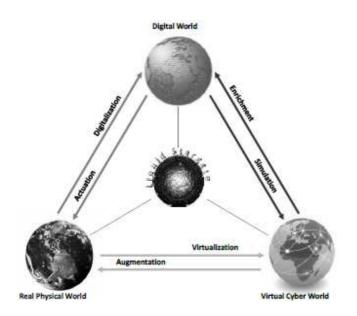


Fig. 2: OSMOSE Liquid Sensing Enterprise Framework [8]

Following the OSMOSE metaphor, each IoT object system has a physical presence in the real world, a model in the digital world specifying predefined patterns or behaviours, and an image in the virtual world to project hypothetical what-if scenarios ruled by some laws and policies (e.g. forecasting the behaviour of a car by mathematical models, or simulating the performance of a production plant given some dynamic inputs, or modelling crowd behaviour in panic situations at the stadium). Thus, from the OSMOSE perspective, the three worlds can be represented as a set of enterprise processes and sub-processes, with their internal tasks and events. The events can be generated in any of the three worlds of a liquid sensing enterprise and a set of them (osmosis events) can be propagated to the other two worlds by triggering the so called 'Osmosis Processes'. A triangular diagram (Fig. 2) can be used to represent these processes used to manage the interactions between the worlds.

Nevertheless, there should be some automatic, real-time but context-based mechanism for consistency keeping and reconciliation of the shadow images. Thus, regular interoperability techniques for exchanging information should be substituted by an integrated intelligent entities visible through the gates of each world (as in the 1994 Stargate movie teleportation device - imdb.com/title/tt0111282/).

#### 2.2. Osmosis Processes

The osmosis processes are a special type of processes used to moderate the information exchanged among the worlds. As mentioned in the previous sub-section, in the OSMOSE context are distinguished six osmosis processes [4]: 1) Digitalization: Modelling and representation of real world data in a computer-tractable form; integration of event driven with service-oriented architecture; 2) Actuation: Plan and implement highly distributed decision-making. Takes processes through pervasive and autonomous networks of smart objects and intelligent things; 3) Enrichment: Extends the computational and experiential capabilities of the digital world with annotations and projections coming from simulations and what-if hypothetical scenarios; 4) Simulation: Instantiate and run hypothetical future scenarios fed by digital world data; 5) Virtualisation: Provides data for simulation of hypothetical simulations from the real world and runs the simulation; and 6) Augmentation: Annotates real world objects with virtual world information;

### **3. Event Driven Architectures**

Especially with the advent of the Internet of things (IoT) and big data, interest in event driven architectures (EDA) grew significantly. EDA and service-oriented architectures (SOA) complement each other [9]. While SOA is synchronous and mainly covers processes which run according to their design, EDA is inherently asynchronous. Combining the two is a complex endeavour and generally applicable concepts are therefore either rather too unspecific or too difficult to apply in different application contexts. Rather than trying to give general advise for the design of such applications we present how these concepts were used in the OSMOSE project.

#### **3.1 Event Processing**

Event processing is the basis of an EDA. "An event is an occurrence within an particular system or domain; it is something that has happened, or is contemplated as having happened in that domain. The word event is also used to mean a programming entity that represents such an occurrence in a computing system" [10]. According to Michelson there are three general styles of event processing [9]:

1. Simple Event Processing: An event happens which triggers downstream actions;

2. **Stream Event Processing**: Is used to process the real-time flow of information in and around the enterprise which enables in-time decision making;

3. **Complex Event Processing**: Evaluates a confluence of events to derive actions to take. Events may cross event types and occur over a long period of time.

In a mature EDA the three basic styles are most of the time used together. OSMOSE's aim is not to contribute to the investigation of the general concepts or to come up with a new execution machinery but rather to define from a methodological point of view how to deal with the existing concepts and tools in an osmosis architecture.

The main interest of OSMOSE is to investigate osmosis processes where events are passed between the real, digital, and virtual worlds. In the first place this means that event processing needs to take place in all three worlds and events have to be classified according to whether they trigger osmosis processes, i.e. processes which passes the

events from one world to another, possibly with additional information which is referenced in the event. In OSMOSE we do not try to come up with a new event ontology or event classification but rather build on what already is there. We therefore only define a simple envelope to pass on information between the different worlds. The following example shows an instance of such an event envelope:

<domain> Aeronautics </domain> <worldID> Real World </worldID> <eventType> Critical Error </eventType> <timeStamp> Time </timeStamp> <eventResourceURI> URI </eventResourceURI> <dataURI> URI </dataURI>

In this envelope the attribute eventType needs to be further clarified. We combined the ontologies from the BPMN 2.0 event package and linking open descriptions of events (LODE) [11] to classify and describe the events in the three different worlds. The idea of the envelope is that only the event type and the resource which was the source of the event is provided. All the rest of the information is provided in the document the URI in the data attribute is referring to. The osmosis processes which transfer the events from one world into another one are themselves described in BPMN 2.0 [12].

#### 3.2 Context Awareness

According to Dey and Abowd [13], context can be described as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". In the OSMOSE Project, shadow images of entities present in all of the worlds need to be kept consistent. Thus, managing the context of entities is extremely important. Knowledge about the context needs to be structured and provided in a machine-readable way. Ontologies have raised attention in knowledge management systems over the past decade [14]. They can be useful to represent knowledge formally and to enable shared knowledge bases, from which knowledge can be inferred. Ontology modularization [15] divides large monolithic ontologies into domain modules, such as entity, process, event or service ontologies.

Each OSMOSE world structures information in its own ontology module, called *knowledge base extension* (KBE). Thus, the three separated worlds are enabled to handle relevant context information independently, adhering to the concepts given by the liquid sensing enterprise. Background consistency of contextual information is enabled by inter-world communication with the six osmosis processes.

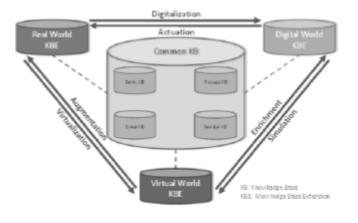


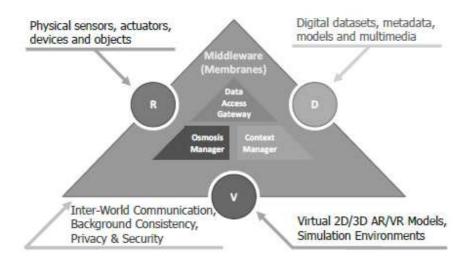
Fig. 3: OSMOSE Metamodel

# 4. A Conceptual Architecture for the OSMOSE LSE

The conceptual architecture follows the OSMOSE metaphor, in which the three worlds, namely the real, digital and virtual world, are separated from each other and bound by a semi-permeable membrane. The overall OSMOSE architecture is illustrated in Fig. 4. Communication between the three worlds has to pass the membrane between the worlds, which is incorporated in the OSMOSE middleware.

The OSMOSE middleware is responsible for intelligent communication delegation which is supported by osmosis and context management. A **data access gateway** allows the **liquid stargate** to access data and models of the three worlds seamlessly.

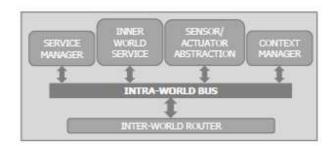
By utilizing complex event processing and Semantic Web technologies, intelligent and controlled communication is enabled.



#### Fig. 4: Overall OSMOSE Architecture

#### 4.1. OSMOSE Worlds Architecture

The OSMOSE world template serves as a generic architecture for each of the OSMOSE worlds that are connected to the OSMOSE Middleware. The **inter-world router** acts as a gateway and interface for inter-world communication, thus taking on the role of the semi-permeable membrane concept. Messages or events from inside the worlds that are passed to this component enriched with necessary information for inter-world routing and delegated to the OSMOSE middleware. Messages or events from outside the worlds are unwrapped and delegated to the intra-world bus utilizing complex event processing and Semantic Web technologies. Services inside the worlds are registered at the inner enterprise service bus.



#### Fig. 5: OSMOSE World Template

In IoT systems, the abstraction of physical sensors, actuators, devices and objects play a fundamental role, allowing end users or applications to access high-level representations of IoT information, regardless of the underlying implementation. The OSMOSE real world provides inner-world services for devices and objects from the physical world which serve as a high-level abstraction in order to overcome heterogeneity. Devices and objects are deployed as services and registered at the service manager, which orchestrates the communication flow over the real world bus. The OSMOSE virtual world provides the users or applications with services to access and interact with virtual reality (VR) and augmented reality (AR) models such as 3D models of physical objects, augmented reality information or virtual simulators in an integrated manner. The OSMOSE digital world incorporates services for user models and multimedia representation of data. OSMOSE applications are able to adapt to user behaviour, thus making it crucial for the OSMOSE system to handle user context. User context and information is handled inside the **user content service** which allows applications to access this information and adapt to the user. Multimedia representations are handled inside the **multimedia content service**. The described worlds are instantiations of the OSMOSE world template connected to the inter-world router of the middleware described in the following section.

#### 4.2. OSMOSE Middleware Architecture

Like in the OSMOSE worlds, the **inter-world router** acts as an endpoint to the inner workings. The OSMOSE worlds are connected to this endpoint by their own inter-world router. Messages or events coming from the real, virtual or digital world are unwrapped and delegated to the **membrane bus**. The **membrane bus** handles further communication between the OSMOSE middleware components.

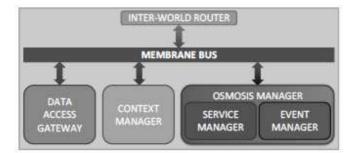


Fig. 6: OSMOSE Middleware Architecture

The event manager handles events processing of the OSMOSE middleware services. Events are sensed in order to provide classification and intelligent communication. The service manager handles service orchestration and choreography of the OSMOSE middleware services. The context manager comprises contextual information about entities, events, services and processes that are related to the OSMOSE worlds and regarding inter-world communication. Thus, the context manager incorporates all necessary information to distinguish between events leading to osmosis processes. The data access gateway provides interfaces for sensing-liquid enterprise applications and the liquid stargate. The communication inside the OSMOSE middleware is enabled via the membrane bus.

# 5. Deployed Middleware and Implementation

The middleware architecture aims to support the execution of osmosis processes. In particular the middleware refers to the management of the OSMOSE membrane that should monitor events and allow the exchange of information among the three different worlds at the time an osmosis event is recognized and pass through the membrane. Thus, the membrane process (see OSMOSE lane of Fig. 7) acts as a listener component that, at the time events are received, checks their context and nature (*eventType*) and decides whether an event is an osmosis event or not. If it is the case, the corresponding osmosis process(es) is/are started. Otherwise the system can decide to do nothing or to start an intra-world process needed to manage the event in the scope of the world that originated it.

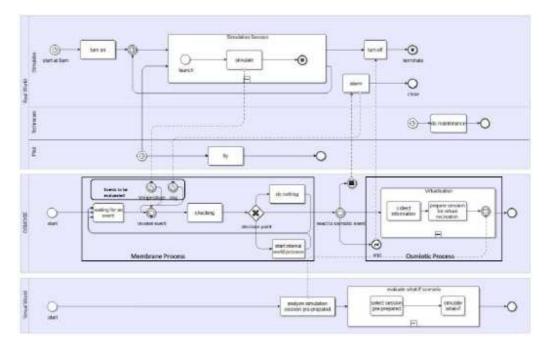


Fig. 7: Example of recognition, generation and management of an osmosis event

Fig. 7 provides an example for the recognition, generation and management of an osmosis event. The use case is based on the aeronautics industry scenario in which helicopter training solutions are provided. Flight simulators are intended as training device which let pilots perform several types of training in different flight scenarios. The most important feature of these devices is the possibility to reproduce system malfunction and emergency scenario that for obvious reasons cannot be done on the real aircraft. These flight simulators can lead to a higher level of training if compared to the real helicopter, thanks to the several flight scenarios that can be reproduced on this kind of device. However, the Flight Simulator itself is not free of bugs and maintenance activities. Thus, the presented example is triggered following an identification of a situation that can be dangerous for the safety of pilots and the simulator itself, because a simulator component is overheated.

**Events generation** – the hardware simulator (seen as a business process in the real world) has a sub-process that is the simulation session. During each session, events (signal) are sent to the middleware, and in particular the temperatureSignal is monitored.

**Event evaluation** – the membrane process contains the rules to evaluate the type of the event. Here the event (or a set/sequence of events or a pattern) is recognised as an osmosis event because certain conditions are met. In this example, the temperature is over the threshold specified for that context. So it is so recognized as a DANGER (osmosis) event. In case it is not recognised as osmosis event the monitor can decide to do nothing or to launch a process in the same world from which the event comes.

**Reaction to the osmosis event** – the system reacts to the recognised DANGER event, decomposing the event and executing two services or activities in the originating world to respond accordingly: just warning the maintenance responsible about the temperature of a specific hardware component (e.g. email since the situation is not severe); or shut down the simulator and send a mobile phone message to the maintenance (it requires immediate attention). In parallel, it starts the virtualisation osmosis process.

**Virtualisation process** – the virtualisation process aims at preparing the data for launching a what-if scenario. The process collects data about the real world flight simulator and flight simulation session, as well as components temperature, to create a package that will be used to investigate the problem in a virtual recreation (virtual world). The process ends with a pending virtualisation event to be consumed from the virtual world asynchronously.

**Consuming the virtualisation event** – at the time the human accesses the virtual world, the pending event is loaded and the human is notified to load simulation data and investigate the problem occurred by simulating different scenarios. Possible scenarios to test are "What-if the simulator is overheating because the lamps intensity is too high?" and "What-if the simulator is overheating because some simulator parts are worn down or broken".

For the implementation of the scenario, the temperature value was sent to the Esper<sup>1</sup> complex event processor, with a predefined frequency (e.g. 10 per second). Using the predefined threshold values, Esper rules checked whether there is the danger of overheating of the lamps. Depending on the time pattern of the temperature profile messages events in RabbitMQ<sup>2</sup> were created. This could result in messages (Email or SMS) sent to technicians informing them about the critical status of the lamp and/or in triggering an osmosis process as explained above. The processes were modelled in BPMN 2.0 and executed in the Jadex<sup>3</sup> agent platform.

## 6. Conclusions and Future work

The presented paper addresses the liquid-sensing enterprise paradigm, where exists a need for a convincing unified model and common reference architecture. To address this need, a framework based on the osmosis metaphor has been proposed on the OSMOSE project together with the corresponding architecture. In this architecture the communication between the three worlds (real world, digital world, virtual world) passes through the osmosis membrane, which is incorporated in the OSMOSE middleware architecture. It monitors the events and allows the exchange of information among the worlds. To demonstrate the successful application of the architecture, it was deployed and implemented in accordance to one of the project pilots (aeronautics).

The major next steps towards the final architecture approach comprise functional and non-functional evaluation and testing in order to validate and verify the integrated architecture. Proper testing procedures will be applied to assess the compliance of the whole system with respect to the original user specifications. Additionally, osmosis processes modelling, performance, and availability to complex event processing tools need to be further developed, analysed, and evaluated.

# Acknowledgments

Authors would like to acknowledge the European funded Project OSMOSE (FP7 610905) that supported the development of various ideas, concepts and use case presented.

# References

- [1] FInES Research Roadmap Force, "FInES Research Roadmap 2015," 2012. [Online]. Available: http://cordis.europa.eu/fp7/ict/enet/documents/fines-research-roadmap-v30\_en.pdf. [Accessed: 01-Jan-2015].
- [2] O. Vermesan and P. Friess, Eds., Internet of Things: From Research and Innovation to Market Deployment. Aalborg: River, 2014.
- [3] R. Jardim-Gonçalves, A. Grilo, A. Panneto, and H. Molina, "Invited Session on Digital and Sensing Enterprise," in 19th IFAC World Congress, 2014.
- [4] C. Agostinho, M. Sesana, R. Jardim-Gonçalves, and S. Gusmeroli, "Model-Driven Service Engineering Towards the Manufacturing Liquid-Sensing Enterprise," in 4th International Cce on Model-Driven and Software Development, 2015.

<sup>&</sup>lt;sup>1</sup> http://www.espertech.com/esper/index\_redirected.php

<sup>2</sup> https://www.rabbitmq.com/

<sup>3</sup> http://www.activecomponents.org/bin/view/About/New+Home

- [5] C. Agostinho and R. Jardim-Gonçalves, "Sustaining interoperability of networked liquid-sensing enterprises: A complex systems perspective," Annu. Rev. Control, 2015.
- [6] ISA Interoperability Solutions for European Public Administrations, "European Interoperability Framework (EIF) for European public services," 2010. [Online]. Available: http://ec.europa.eu/isa/documents/isa annex ii eif en.pdf. [Accessed: 11-Jan-2015].
- [7] European Research Cluster On The Internet Of Things, "Internet of Things: Pan European Research and Innovation Vision," 2011.
- [8] E. Marcelino-Jesus, J. Sarraipa, and C. Agostinho, "A Requirements Engineering Methodology for Technological Innovation Assessment," in Moving Integrated Product Development to Service Clouds in the Global Economy, 2014, pp. 577–586.
- [9] B. M. Michelson, "Event-Driven Architecture Overview," Patricia Seybold Group, Feb, 2006.
- [10] O. Etzion and P. Niblett, Event Processing in Action. Manning Publications Company, 2010.
- [11] R. Shaw, R. Troncy, and L. Hardman, "LODE: Linking Open Descriptions of Events," in Proceedings of the 4th Asian Conference on The Semantic Web, 2009, pp. 153–167.
- [12] C. Natschläger, "Towards a BPMN 2.0 Ontology," in Business Process Model and Notation, vol. 95, R. Dijkman, J. Hofstetter, and J. Koehler, Eds. Springer Berlin Heidelberg, 2011, pp. 1–15.
- [13] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, 1999, pp. 304–307.
- [14] M. a. Ale, O. Chiotti, and M. R. Galli, "A Distributed Knowledge Management Conceptual Model for Knowledge Organizations," ICFAI J. Knowl. Manag., vol. 3, pp. 27–39, 2005.
- [15] S. Ben Abbès, A. Scheuermann, T. Meilender, and M. d'Aquin, "Characterizing Modular Ontologies," in Proceedings of the 6th International Workshop on Modular Ontologies, Graz, Austria, July 24, 2012, 2012, vol. 875.