

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/322100557>

Enterprise operating system framework: federated interoperability based on HLA

Article in *International Journal of Simulation and Process Modelling* · April 2018

DOI: 10.1504/IJSPM.2018.10014989

CITATIONS

0

4 authors:



Gregory Zacharewicz

University of Bordeaux

125 PUBLICATIONS **500** CITATIONS

[SEE PROFILE](#)



Joseph Youssef

Université Bordeaux 1

6 PUBLICATIONS **9** CITATIONS

[SEE PROFILE](#)



David Chen

Shih Hsin University

117 PUBLICATIONS **2,157** CITATIONS

[SEE PROFILE](#)



Zhiying Tu

Harbin Institute of Technology

15 PUBLICATIONS **36** CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Distributed Simulation for "Greener" Transportation of Smart Product [View project](#)



Multi Agent/HLA Enterprise Interoperability [View project](#)

Enterprise operating system framework: federated interoperability based on HLA

Joseph Rahme Youssef*, Gregory Zacharewicz
and David Chen

Laboratoire IMS-LAPS UMR
CNRS 5218 Groupe Productique (GRAI),
351 Cours de la Libération,
33405 Talence cedex, France and University of Bordeaux,
351, Cours de la libération,
33405 Talence cedex, France
Email: joseph.youssef@ims-bordeaux.fr
Email: gregory.zacharewicz@ims-bordeaux.fr
Email: david.chen@ims-bordeaux.fr
*Corresponding author

Zhiying Tu

School of Software,
Harbin Institute of Technology,
92 West Dazhi Street, Nan Gang District,
Harbin, China
Email: tzy_hit@hit.edu.cn

Abstract: Operating system (OS) is a well-known concept in computer science as an interface between human and computer hardware (MacOS, Windows, IOS, Android). In the perspective of developing future generation of enterprise systems based on IoT and cyber-physical system principles, this paper proposes to develop an enterprise operating system (EOS) that will act as an interface between enterprise business managers and enterprise resources performing enterprise operations. The paper presents at first a set of requirements and functionalities of EOS. Then a survey on existing relevant works is given and mapped to the requirements. After that the existing models related to the federated enterprise interoperability are presented. The architectures of envisioned EOS and the federated interoperability are outlined. The last part draws some conclusions and gives future perspectives.

Keywords: operating system; OS; architecture; model; infrastructure; interoperability.

Reference to this paper should be made as follows: Youssef, J.R., Zacharewicz, G., Chen, D. and Tu, Z. (xxxx) 'Enterprise operating system framework: federated interoperability based on HLA', *Int. J. Simulation and Process Modelling*, Vol. X, No. Y, pp.xxx-xxx.

Biographical notes:

This paper is a revised and expanded version of a paper entitled 'Developing an enterprise operating system (EOS) with the federated interoperability approach' presented at 13th International Multidisciplinary Modeling & Simulation Multiconference – I3M, Larnaca, September 2016.

1 Introduction

Today, to be competitive in the market, enterprise operations need to be efficiently controlled and coordinated in order to react quickly to constantly changing market demands. To reach this goal, enterprise operations must be in real-time monitored and its resources be integrated as one system.

In the past, considerable efforts have been made to develop enterprise integration, enterprise resource planning (ERP) and management solutions; however those approaches fail to provide satisfactory solutions, especially for SMEs. Enterprise integration is still seen as a complex, costly and time-consuming task.

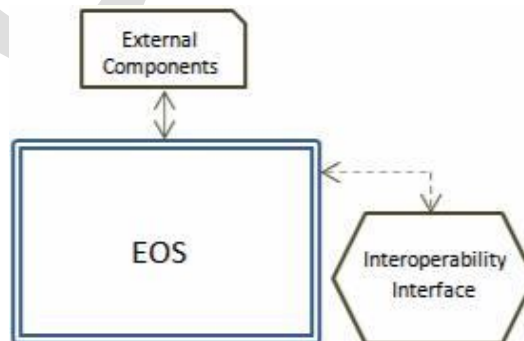
In parallel to the R&D on enterprise integration, ERP solutions have been massively implemented in enterprises, especially in large and medium size companies. During the last couple of years, the image of ERP systems seems to have changed from a highly promising into a very demanding system (Burnson, 2015). Most of existing ERP solutions in the market is proprietary systems. The interoperability between an ERP system and other systems in a company reminds difficult. Moreover, introducing an ERP solution in a company is becoming increasingly in size, rigidity and complexity which make the implementation and the updates very time consuming, involving high costs and disrupting current processes and operations.

In line with the development of the future generation enterprise manufacturing systems based on IoT and cyber physical system principle, this paper tentatively presents the latest development of an enterprise operating system (EOS). It aims at setting a loose coupled connection between enterprise's software applications through the federated interoperability principle. EOS allows joining existing relevant approaches together for complementarities and synergies (Burnson, 2015). The cooperating parties can accommodate and adjust 'on-the-fly' to ensure quick interoperability establishment, easy-pass, and dynamic environment update. The envisioned EOS will behave as an OS of the computer to monitor enterprise resources and coordinate enterprise operations.

The approach presented in the paper is part of an ongoing research activity carried out at IMS laboratory of University of Bordeaux to develop enterprise interoperability in order to improve the performance and competitiveness of small and medium size manufacturing enterprises in collaborative environments by executing enterprise models. In the paper focus is given on the EOS concepts and principles, the requirements and conceptual architecture of EOS as well as the technical and implementation architectures based on a federated interoperability framework.

The proposed EOS will behave in the same way as an operating system (OS) of the computer, but under enterprise context. It will act as a system-wide interface between enterprise business managers and enterprise resources performing daily enterprise operations (see Figure 1). The EOS will manage and interpret the enterprise content and requests defined by business managers and activities owner, trigger various enterprise operations with dynamically allocated enterprise resources (human, machining, computing), control the processes execution, and monitor the status of enterprise resources through various sensing devices and front-ends (Chen et al., 2015; Zhiying et al., 2012).

Figure 1 EOS, interoperability interface and the external peripherals (see online version for colours)



To make this flexible, dynamically configured and real-time controlled enterprise a reality, a sound EOS capable of (real-time) monitoring enterprise resources and operations in order to dynamically allocate resources to required activities is more than a necessity but a pre-condition. In this context, the proposed EOS will be implemented as a new virtual back-end platform that allows the dynamic allocation of the enterprise activities with its related resources for ensuring the flexibility and real-time controlling of enterprise operations. This EOS will connect and combine the heterogeneous sensors and systems from multi-vendors in order to work as 'one' configured system.

One of our motivations to propose EOS is due to the fact that today, enterprise operational management is largely dominated by traditional holistic enterprise management solutions like ERP which is a top-down integrated approach that includes in its framework the main enterprise applications and operating functions (Burnson, 2015). Nevertheless, this solution may constraint the business users due to the top-down 'enclosing' paradigm. It is difficult for an enterprise to implement its tailored solutions according to the specific needs of each particular enterprise.

An alternative approach would be to provide loosely coupled connections between enterprise's software applications (federated interoperability) with the support of only one 'core central orchestrator' (EOS).

Another reason of proposing the EOS is to develop a response to the foreseen enterprise evolution towards a new generation of modular structured agile 'smart factory'. The proposed EOS will support this Fourth Industrial Revolution (or Industry 4.0) with federated interoperability and IoT.

In other words, EOS is a bottom-up federated approach only providing EOS (functions) that allows various heterogeneous applications from different vendors to connect to EOS and to work together.

From this perspective, EOS's approach presents the better perspective for industry in the context of factory of future (FoF). It allows connectivity, flexibility and real-time control with a bottom-up approach. It also allows providing more options for those enterprises with tailored solution, especially for SMEs. With a lower cost comparing with ERP, an enterprise equipped with an EOS will only deploy and install its required heterogeneous applications. Once EOS adopted in industry with the federated interoperability approach, a foreseen ecosystem would be developed in consequence to provide varieties of enterprise applications compatible to EOS.

2 Existing works relevant to EOS

Since the beginning of 80's, several architectures, platforms and models have been developed which are relevant in developing EOS. Eight approaches were identified and are reviewed in this section.

2.1 Existing works

The common object request broker architecture 'CORBA' was designed for facilitating and enabling the communication of systems deploying on diverse platforms, written in different languages and implemented on different OSs and models. It offers benefits such as inheritance, information hiding, reusability and polymorphism. However, CORBA is seen more as an enterprise application integration (EAI) platform rather than an EOS (Gray and Reuter, 1993).

ENV 13550 was developed in 1990's to support the computer integrated manufacturing 'CIM'. The main objective is to support system-wide business process monitoring and control (Leadbeater, 1999). ENV 13550 defines the requirements to express the capabilities of environments for developing, executing and integrating enterprise models on an open IT based platform. ENV 13550 focused on enterprise model execution; integration of divers' heterogeneous applications is not well supported as CORBA (Shorter, 1997; Zacharewicz et al., 2016). Interoperability issues were not addressed in ENV 13550.

Another well-known work similar to CORBA is EAI that provides integration of systems and applications residing on different OSs and using different database solutions across an enterprise (SCM, ERP, CRM, payroll...) (AMICE, 1993). As CORBA, EAI was not dedicated to enterprise process and operation monitoring and control that are to be supported by EOS. It aims at integration approach rather than loosely coupled interoperability (Linthicum, 1999).

Concerning this last point, we must mention the workflow management coalition 'WfMC' which focuses principally around process definition file interchange by using the standard 'XPDL' which presents a process design format for storing the visual diagram and process syntax of business process models (Hollingsworth, 1994). Like ENV 13550, 'WfMC' aims at process model execution that is one of the main functions in EOS. It is worth mentioning two other early works to support integration and interoperability of heterogeneous enterprise systems: OSF/DCE and ODP. OSF/DCE is used to allow different OSs to distribute processing and data across the enterprise. It supports the sharing of information and provides services to mask the complexity of multi-vendor network environments (Turner, 2012). ODP describes and builds widely distributed systems and applications in a multi-vendor environment (Zurawski, 2004). Both ODP and DCE are similar approaches aiming at supporting system-wide distributed communication and sharing of information that is also a core functionality to be provided by EOS.

Further to the above-mentioned works, service orientation has gained interest in the middle of 90's and has been used as a basic principle to develop enterprise integration and interoperability platforms. To this end, the most well-known are enterprise service bus (ESB) and service oriented architecture (SOA). The 'ESB' is developed in analogy to the bus concept found in computer hardware architecture combined with the high-performance

computer OSs, and used for designing and implementing communication between mutually interacting computing hardware (Flurry and Clark, 2011). While the 'SOA' is an architectural style that support service orientation including service-based development and outcomes services (Salayandia and Gates, 2007). This technique involves the interaction between loosely coupled services that function independently and enables networked applications to cooperate and exchange information. It does not directly support functions required by an EOS but providing interesting principle to develop the EOS.

Both ESB and SOA cannot directly be used as an EOS but they provide interesting concepts and principle to develop and build the EOS (The Open Group, 2012).

2.2 Efficiency of existing techniques for EOS

Considering enterprise resource management (ERM), ENV 13550 and to a lesser extent the WfMC deal with the human and machine resources; the other existing approaches only focus on IT resource management specially monitoring and control. SOA allows to leverage existing resources and integrate them using service-oriented integration techniques. It delivers the flexibility and agility that business resources require in order to facilitate the combined use of the resources to deliver various required services; this is a useful approach for EOS (AIIM, 2000; Bieberstein et al., 2005; Favre, 2003; Moon and Lee, 2006; Rademakers and Dirksen, 2008; Zaraté, 2013).

As for the enterprise process management (EPM), most of the existing approaches include this functionality; However, in an EOS, the 'business process' defined by business users and the 'process' that is orchestrated to ensure internal working of platforms, are both included but the focus is on the business process execution. In this sense, ENV 13550 and WfMC provide the most interesting concepts and principles to define EOS by controlling the process model run-time execution (sending command triggering the starting of processes, recording the ending status and sending required capability to resource management to find/allocate the available qualified

resource) (Chang, 2005; Fagg et al., 1996; Gable, 2002; Henning, 2006; Sethi et al., 1995; Vollmer, 2011).

Referring to enterprise information management (EIM), all existing approach support the intercommunication of information between the EOS internal and external components which is an essential function; However due to the heterogeneous data sources in an enterprise, interoperability is fundamental to support this function (Rasta, 2013; Wainer et al., 2000).

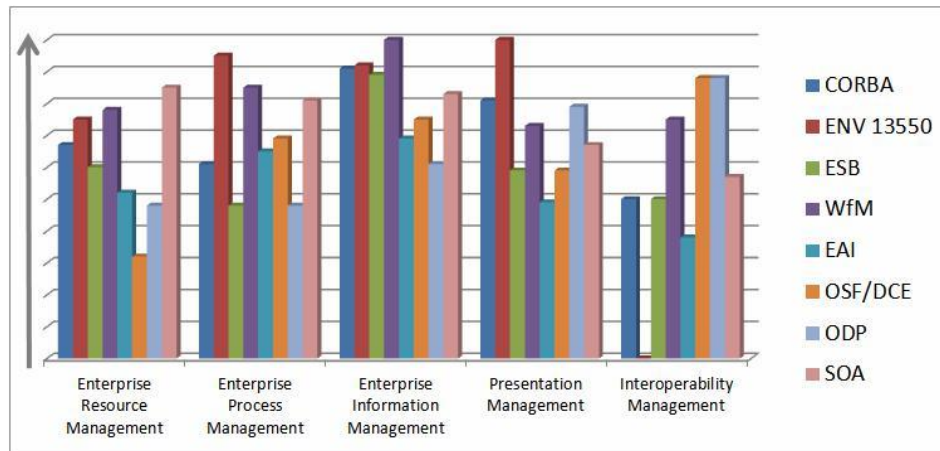
Concerning the presentation management (PM), this functionality is necessary to interact with all resources and business users connected to an EOS, ENV 13550 provides a set of interfaces and commands between internal and external worlds of the platform. It organises the information flow between the platform and the enterprise resources, and interacts with the human, machine and IT dialogue services (CEITON, 2014; Jin, 2009; Wyszowski, 2011).

For interoperability management (IM), existing approaches did not develop this function to a mature and satisfactory level. They are focusing on the enterprise integration and none of them proposes the complete solution for all the interoperability issues especially to ensure the loosely coupled interoperability. A set of interoperability utility services needs to be developed to support the use of the EOS (Konstantas et al., 2006; May, 2001; Omicini et al., 2013; Saha et al., 2011; Xu, 2014).

In conclusion, the existing works are all relevant to designing and developing an EOS but they do not support and cover all the requirements and functionalities required by EOS. Each of them has its particular objectives with specific focuses. CORBA and EAI are seen more as an EAI platform rather than an EOS. ENV 13550 moved a step towards an EOS, but unfortunately the work of ENV 13550 is discontinued. The WfMC, ODP and OSF ensure the information management by executing the intercommunication data exchange, but they do not cover all the EOS functionalities. The ESB and SOA developed to ensure enterprise integration and interoperability platforms, cannot be directly used as an EOS but they provide interesting concepts and principle to develop and build the EOS.

Table 1 Comparison of existing approaches against EOS requirements

	<i>Enterprise resource management</i>	<i>Enterprise process management</i>	<i>Enterprise information management</i>	<i>Presentation management</i>	<i>Interoperability management</i>
CORBA	+	+	+++	+++	+
ENV 13550	+++	+++	+++	+++	—
ESB	++	+	++	—	—
WfM	++	++	+++	+++	++
EAI	++	++	+++	+	+
OSF/DCE	—	+	+++	+++	++
ODP	+	—	+++	+++	++
SOA	+	++	++	++	+

Figure 2 Summary of the existing techniques services and efficiencies based on the EOS requirements (see online version for colours)

The following notations are used: ‘+ + +’ means that there is a strong correlation and that the approach highly fulfils the EOS criteria, ‘+’ denotes that there is weak correlation and ‘+ +’ is in between, ‘-’ means that the approach does not meet the criteria. Table 1 gives an evaluation on the degree of relevant of the existing approaches to EOS.

The results of the comparison have been quantified and summarised in Figure 2.

3 Requirements and functionalities of EOS

This section presents a preliminary set of requirements and functionalities for developing an enterprise operating system. They are inspired from and based on the work specified in a pre-European standard known as prENV 13550 ‘Advanced manufacturing technology – systems architecture – enterprise model execution and integration services’ (CEN, 1999; Chen et al., 2015). This pre-European standard benefits from the result of European esprit project CIMOSA (Tolk and Muguira, 2003).

3.1 Enterprise resource management

ERM is the efficient and effective deployment and allocation of an organisation’s resources including: the sets of individuals who make up the workforce of an organisation and business sector ‘human’, the types of devices and tools containing one or more parts that use energy to perform an intended action ‘machines’, and the collections of physical elements that constitutes computer and communications systems as well as software applications ‘IT resources’. ERM is essential for EOS, it provides a real-time and global view of the ‘occupation’ of the resources in a company by monitor the enterprise resources system-wide (available, occupied, out-of-order...) (Gray and Reuter, 1993). Main required functionalities of ERM are: checking the availability of the resources and pre-assigning it when available; signalling the process

management that those critical resources are reserved; matching the required capabilities to the capabilities of existing available resources; managing time and performance management support of enterprise functions and operations; and ensuring that the right resources are allocated to the right place at the right time.

Noting that only the resources connected to EOS is to be monitored and managed by EOS, for example those ones involved supporting defined processes. IT and automated machine resources are to be connected directly to EOS. Human resources are connected to EOS via appropriate portable interfaces.

3.2 Enterprise process management

EPM is required for EOS to help orchestrating and carrying out organisational activities so that operational processes defined by the decision makers are executed, aligned and integrated. Resources required to execute processes are dynamically allocated/de-allocated by ERM. Main functionalities of EPM include: registering and managing process priorities using the run-time repository service; sending commands triggering the starting of processes, recording ending status of processes (done, not done, fail...); responding to rule interpretation notifications when a process is notified as terminated, requesting ERM to take appropriate house-keeping action and invoking events handling to terminate; checking authorisation to execute a process; managing event queues and traceability by time-stamping and recording the history of events; escalating exception-related processes for persistent and non-persistent errors and damage control (Leadbeater, 1999).

3.3 Enterprise information management

EIM supports information and data exchange between all entities connected to the EOS such enterprise resources and business managers (Leadbeater, 1999). EIM is specialised in

finding solutions for optimal use of information within the organisation, for instance to support decision-making processes or day-to-day operations that require the availability of data and knowledge. Main required functionalities of EIM are: maintaining consistency with the system-wide run-time name space; supporting transparent access to system and user data stored in several heterogeneous database systems; supporting transparent access to data with no concern for data location, data replication and storage mean type; and ensuring information, data confidentiality and security to protect from non-authorised access.

3.4 Presentation management

PM is concerned with the interface between internal and external worlds of EOS. It produces a user interface which makes it easy (self-explanatory) and efficient to organise and coordinate the communication and information flow between enterprise resources and internal entities of EOS including the human, machine and applications dialogue services (Gray and Reuter, 1993). It is mainly required for: transposing the control and information inputs into the appropriate forms for the involved functional entities and assigned resources; applying object views to extract that part of the information needed by the involved resources.

3.5 Interoperability management

IM is a utility service of EOS presented as a precondition for a successful integration of the business entities of the enterprise involving business process communication, cooperation and coordination, enterprise-wide knowledge sharing as well as enterprise inter-networking, enabled by application and physical components. The federated approach aims at bridging the gap from interoperability concepts to the implementation of interoperable enterprise systems development by allowing quick interoperability establishment, easy-pass, and dynamic environment update in the heterogeneous and multi-partners environment.

4 Models relevant to federated enterprise interoperability

Several models mentioned in this section have achieved some success in developing systems interoperability based on the federated approach. However, none of them proposes the complete solution for all the interoperability issues in order to develop an interoperability interface for ensuring the inter federates communications and the data connection between the components of the EOS and the external peripherals.

4.1 Existing models

Levels of information systems interoperability (LISI) is mainly used to select pragmatic solutions and a transition path for achieving higher states of capability and

interoperability by focusing on technical interoperability and the complexity of interoperations between systems. But LISI model does not address the environmental and organisational issues that contribute to the construction and maintenance of interoperable systems. Organisational interoperability maturity (OIM) can be seen as the evolved LISI model in the context of the layers developed in the command and control support (C2S) Study by extending LISI into the organisational layer (Chen et al., 1998). The database interoperability and inverted-V model is an overall architecture to merge information comprised in heterogeneous data sources into one technically consistent and semantically coherent information space. However, it is only for data but not procedure or architecture (Tolk, 2011). The levels of conceptual interoperability model (LCIM) have been carried out successfully in simulation domain, but the basic premises apply to many complex sets of interoperating systems (Tolk and Muguira, 2003). The system of systems interoperability (SOSI) extends the existing models by adding a focus on programmatic, constructive and operational issues which must be managed across the life cycle (Morris et al., 2004).

The model driven architecture (MDA) approach focuses on portability, interoperability and reusability starting with the well-known and long established idea of separating the specification of the operation of the system from the details of the way the system uses the capabilities of its software execution platform. It contributes on building an interoperable information and communications technology (ICT) model, from enterprise models to technology models. MDA also provides flexibility and adaptability to accommodate changes at a higher abstraction level. However, several studies doubted that MDA will follow the old way of integrated computer-aided software engineering to ruin, to spend 10% effort to generate incomplete and useless code (80% to 90%), but spend 90% effort on struggling in tracing down the rest part to achieve perfection. In addition, the information is losing during the model transformation, such as details of system behaviours (Amber, 2003). The soundness of the model driven interoperability (MDI) methodology has been demonstrated in the current researches, but no full industrial scale validation has been yet achieved. Only some projects have been especially carried to demonstrate these concepts in an industrial real world significant application (Elvesaeter et al., 2007). ADM shows its strong power in obtaining information from the legacy systems. But, many people doubt on the validity of this information for achieving federated enterprise interoperability. Architecture driven model (ADM) met the same model transformation problems as MDA (OMG, 2010).

The remote method invocation (RMI) allows the programmers to write object-oriented programming in which objects on different computers can interact in a distributed network. The distributed interactive simulation (DIS) defines an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly

interactive activities. The aggregate level simulation protocol (ALSP) describes a collection of infrastructure software and protocols for passing the messages between the various participants of a distributed simulation (Massei et al., 2013).

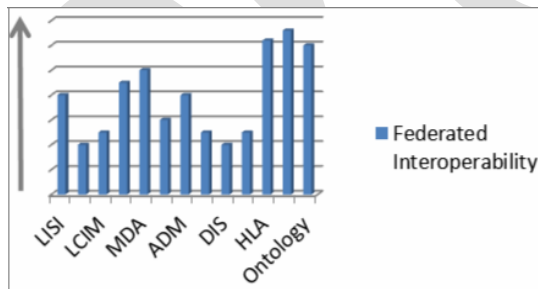
The RMI, DIS and ALSP simulation and application distribution frameworks can support distributed system interoperability, but in varying degrees. None of them can fully satisfy the requirement of the federated approach and especially as concerned with the component coupling, time management, ownership management, environment flexibility and data distribution services (Bruzzone et al., 2005; Buss and Jackson, 1998; IEEE, 1995; Weatherly et al., 1993).

The ontology can fully support the conceptual enterprise interoperability. However, enterprises require more and more dynamic, complex, and advanced interoperability, this kind of architectures independently can hardly handle the updated requirements (Zacharewicz et al., 2009).

4.2 Efficiency of models for federated interoperability concept

Due to the fact that enterprises require more and more dynamic, complex, and advanced interoperability, these methodologies, technologies, and architectures independently can hardly handle these requirements any more. A harmonised and reversible high level architecture (HLA) based methodology is being implemented as a novel idea that combines the existing methods and techniques to develop model driven federated enterprise interoperability. This methodology will creatively combine the excellences of some of these existing methodologies, technologies, and architectures, and propose an innovative way to tackle enterprise interoperability at service and data levels through a federated approach.

Figure 3 Summary of the models' services and efficiencies based on the federated interoperability concept (see online version for colours)



5 Architectures

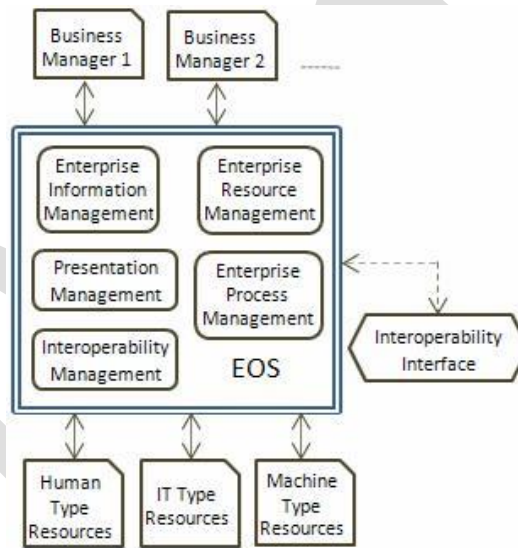
This section outlines the EOS conceptual architecture, EOS technical architecture and the technical architecture of the harmonised/reversible development framework in order to

meet the requirements and functionalities identified in Sections 2 and 3.

5.1 EOS conceptual architecture

Figure 4 gives the EOS conceptual architecture. This EOS is developed and implemented as a new virtual back-end platform that allows the decisions makers to connect to and communicate through the enterprise systems (hardware, software, network, machines...) in an efficient and effective way (Chen et al., 2015).

Figure 4 Eos conceptual architecture (see online version for colours)



Business users and the three types of resources are outside EOS. They are connected to EOS to send and receive information (data, command...).

Human type resources are human operators to be monitored and controlled by EOS. The sets of individuals who make up the workforce of the enterprise. They are commercial and purchasing agents, product designers, production managers, shop floor operators etc...

IT type resources are the physical elements that constitute computer and communications systems. They include computer and other data processing and storage devices, enterprise applications such as MRPII planning software, shop floor scheduling software, CAD, sale forecasting software, CRM software and inventory management software.

Machine type resources are machines, materials and tools containing one or more parts that use energy to perform an intended action. Machines are usually powered by mechanical, chemical, thermal, or electrical means such as automated/manual transfer lines, conventional and NC machines, robots etc.

Business users are not monitored and controlled by EOS; they define what and how enterprise operations will be done and send commands to resources via EOS.

Interoperability interface is the interface that includes the enterprise processes and required information. It enables business interoperability between EOS executed systems and the external components related to the EOS.

As shown in Figure 4, EOS is an interface between business users defining what/how business is to be done, and the three types of enterprise resources performing defined operations.

ERM dynamically monitors the status of enterprise resources, search and allocate suitable resources to operations that must be done, and ensures that the right resources are allocated to the right place at the right time.

EPM executes business processes defined by business users, sends commands triggering the starting of processes, records ending status of processes (done, not done, fail...), and executes EOS internal processes/operations.

EIM manages, protects and supports information and data exchange of all kinds between the enterprise's resources connected to the EOS, and ensures information

and data confidentiality and security to protect from non-authorised access.

PM is a set of services with appropriate interfaces that allow business users and other enterprise resources to connect to EOS and receive/send information.

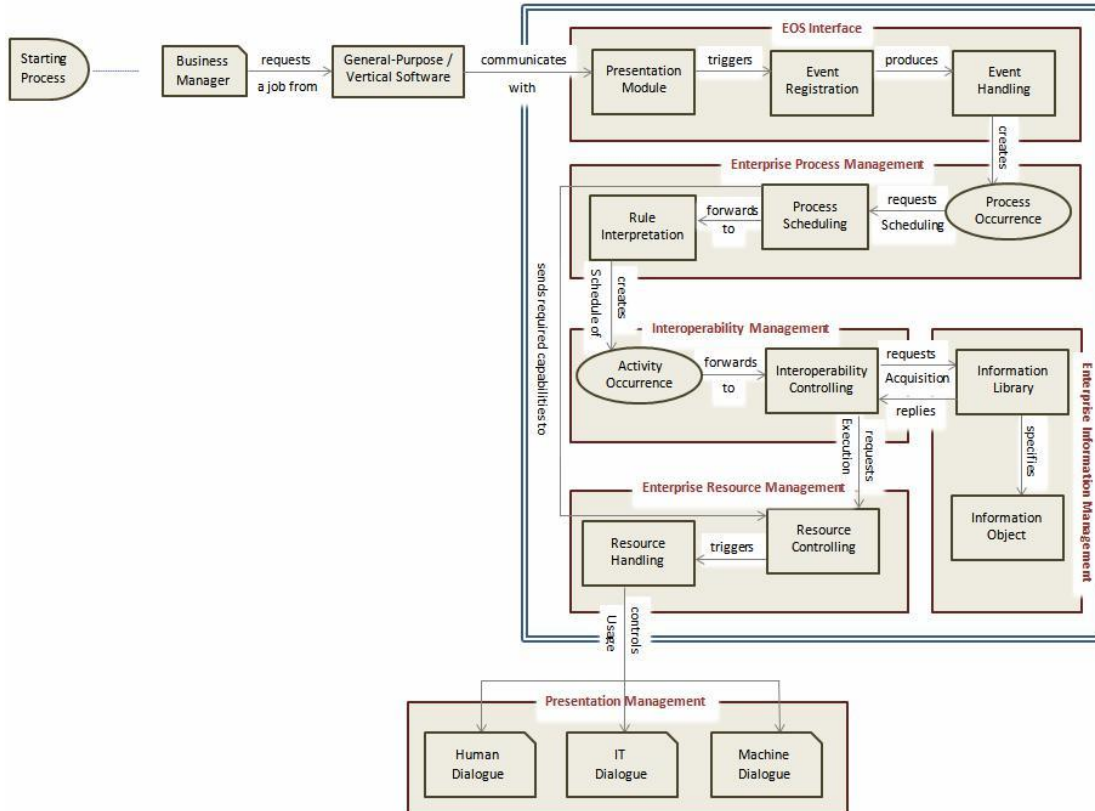
IM is a set of services that provide necessary mapping between heterogeneous resources to make them interoperable through EOS (Youssef et al., 2016b).

5.2 EOS technical architecture

As shown in Figure 5, the enterprise activities are executed and generated through the EOS internal components at beginning from the starting phase.

At first, each business manager accesses the general-purpose and vertical software's interfaces in order to request the day-to-day activities and operations. The related software sends special commands to connect and communicate with the EOS front-end interface called presentation module in order to execute the requested job.

Figure 5 EOS technical architecture (see online version for colours)



After that, the presentation module interprets the run-time entities and triggers the event which will be registered with their associated information through the event registration component using the run-time repository service, and produces the event handling component.

Next, the event handling manages the events' priorities, queues and traceability, provides the order identifier, and creates the process occurrence. The process occurrence manages the process structures and requests scheduling from the process scheduling component which interprets the process behavioural, information and resource requirements. This sub-service checks the authorisation to execute the process, retrieves process descriptions from the run-time repository, invokes the ERM to allocate the required resource capabilities, and forwards the details to the rule interpretation.

Later, the rule interpretation component provides functionality to retrieve the sequencing and conditional rules associated with the identified enterprise process, maintain a state record of all enterprise activities, and respond to detected events in order to initialise or terminate the activity.

Subsequently, the activity occurrence schedule created by the rule interpretation is forwarded to the interoperability component which is responsible of requesting from the ERM to assign resources allocated by the process scheduling component, invoking the EIM to acquire the object states and specify the information object required, requesting from the ERM to release the involved resources when terminating an activity, and signalling the termination of the current activity to the rule interpretation component.

The resource controlling checks the availability of the resources and pre-assigns it when available, responds to the interoperability controlling requests in order to assign agents, and responds to the process scheduling requests to allocate and de-allocate resources. The resource handling component ensures the resource optimisation and the resource mapping by selecting the appropriate resources after matching the capabilities required and by taking into consideration the time, performance and priority.

The PM services are controlled by the resource handling component for handling human, machine and IT dialogues.

The human dialogue provides functionality for presenting in graphical format the current status and the past history of events, allowing authorised personnel to intervene manually in order to modify the contextual parameters at run-time.

The machine dialogue supports the necessary features in order to provide access to the various functional capabilities of the machine. It provides the functionality required for receiving and interpreting responses from the machine.

The IT dialogue provides functionality for interrogating application program interfaces to determine its capabilities, providing support for the integration of the functional entities implemented by existing IT application programs.

5.3 Harmonised and reversible development framework – interoperability interface for EOS

A new framework is presented and will be used as an interoperability interface connected to the EOS in order to set up interoperability rapidly among existing enterprise information systems. This framework will use the existing models benefits for creating a novel way to support the development of federated approach of enterprise interoperability. Thus, the methodology presented will utilise MDA to clarify the system architecture and relationship among systems, and apply model reverse engineering to reuse and align different EOS components and federates to initiate a federate enterprise interoperability environment, and use the HLA and SOA functionalities as technical support.

This framework is mainly used in order to rapidly develop HLA based interface for achieving federated enterprise interoperability.

Primary concepts are separately presented as follow:

- *Harmonised:* HLA is successful in defining how to create a global software execution composed of distributed simulations and software applications. Meanwhile, MDA becomes the standard for promoting the use of models and their transformations to consider and implement different systems. However, they both still have drawbacks expected to be improved. HLA and MDA complement each other by using their achievements, and this harmonisation is conducive for federated enterprise interoperability. This framework is a synthetic framework, which consists of several techniques. As the framework in Figure 6 shows, a new five steps development life cycle is proposed which aligns MDA and HLA FEDEP. MDA is easy to use and understand, and tightly bounded with unified modelling language, meta-object facility (MOF). It appears to be an appropriate choice to overcome the interoperability barriers, which is mentioned in the MDI framework (Elvesaeter et al., 2007). HLA FEDEP is the standard for development and execution of HLA federation. It is quite similar to the waterfall development but with look-back test phase. MDA and HLA FEDEP can be easily aligned, because they have several similar steps. In addition, this framework uses web services to improve the flexibility and compatibility of the HLA (Youssef et al., 2016b). The web services allow potential external systems to discover the existing HLA federation, and then connect to it.
- *Reversible:* means that this framework uses model reverse engineering technique to discover part of the models from the legacy system. Model reverse engineering technique aims at avoiding rebuilding the complete legacy system for a new reuse. The objective is to accelerate the development and reduce the cost. As Figure 6 illustrates, there are two kinds of dotted arrows, which have opposite directions to the five steps development life cycle. These two kinds of arrows represent two different scenarios of model reversal in

this framework. This methodology proposes a way of using UML models discovered from the existing systems to generate HLA FOM that represents system static information and HLA federate code block that represents system behaviour.

- **HLA**: means that this framework dedicates to the development of HLA based application. The RTI used in this approach is an open source RTI, poRTIco. The reason of choosing it is not only because of the software price, but also the objective of initiating a global open framework and receiving comments from contributors who can be interested in this idea. In addition, as mentioned earlier in harmonised part, web services will be used to improve the limitation of the traditional HLA.

This framework creatively combine the excellences of the existing methodologies, technologies, and architectures, and propose an innovative way to tackle enterprise

interoperability at service and data levels through a federated approach.

The goal is to achieve the interoperability among those existing federates in a common project context. The steps of this approach are presented as follows:

- Step 1 (arrows numbered with '1'): model reverse engineering is used to discover the models from the legacy system. The model discovery is guided by the enterprises new requirements and interest. Then, these discovered MDA conceptual models go down again along the alignment of MDA and HLA FEDEP. It means models are generated from code to PSM then PIM and CIM level. At each level of the MDA models the interoperability problem is tracked according to the principle of the MDI framework.

Figure 6 Harmonised and reversible development framework for HLA based application (see online version for colours)

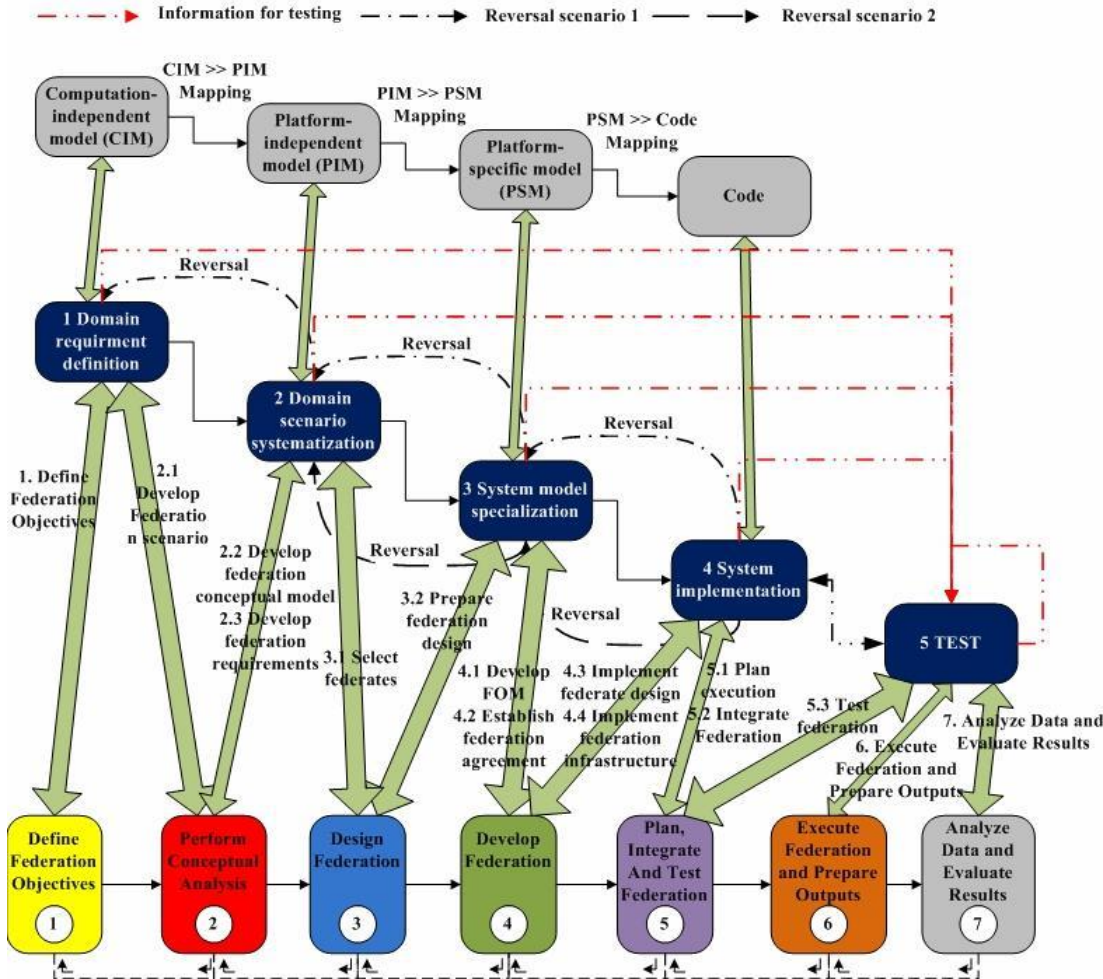


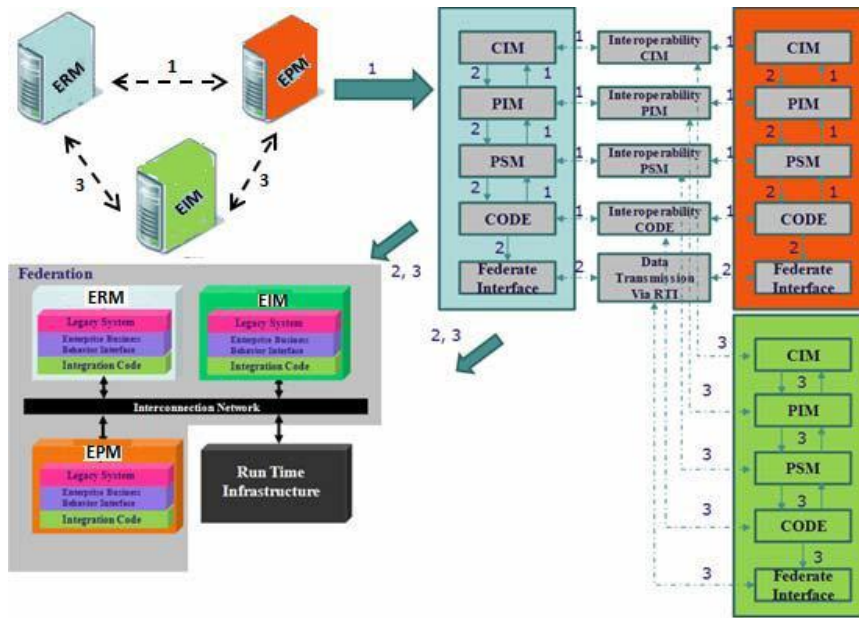
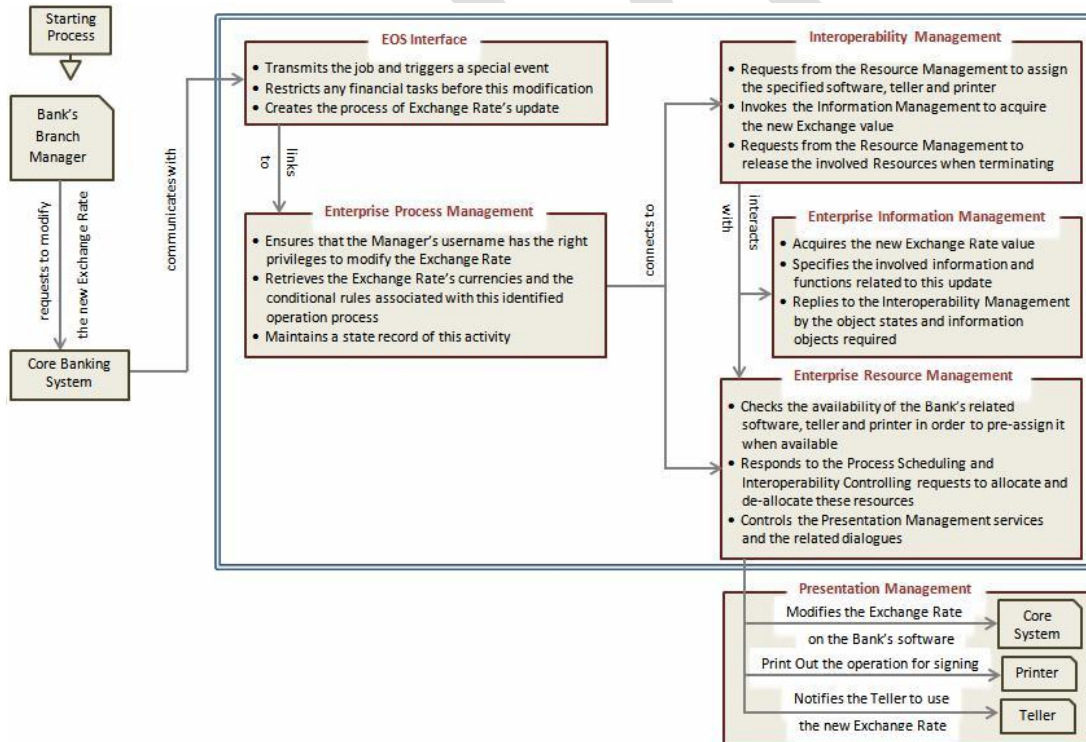
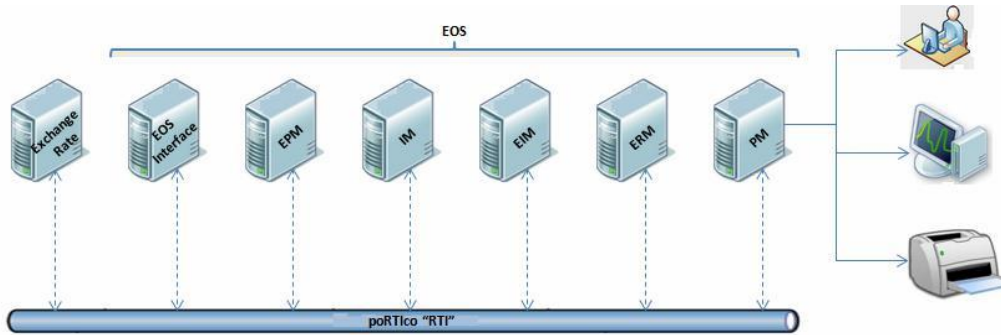
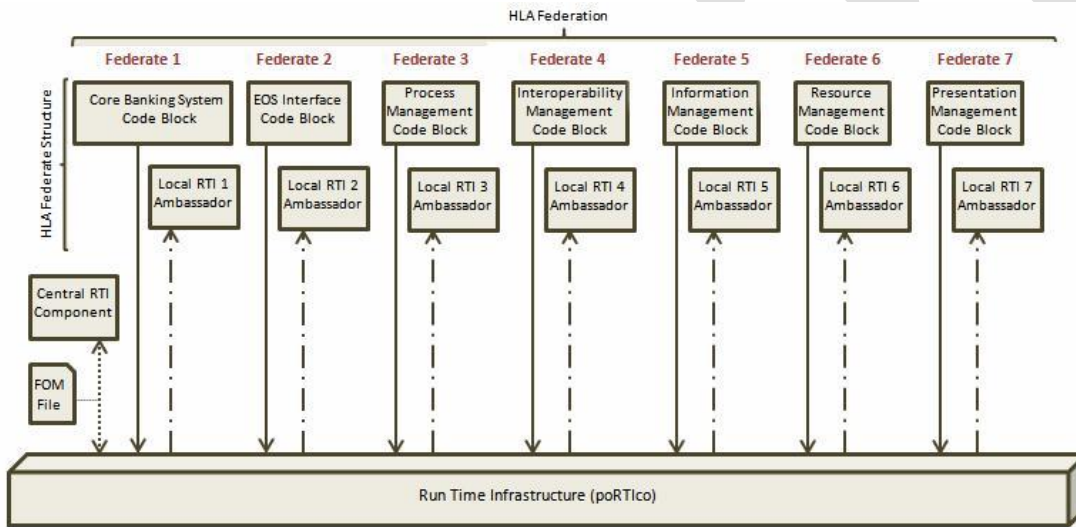
Figure 7 Scenario descriptions (see online version for colours)**Figure 8** Conceptual architecture of the bank's exchange rate update (see online version for colours)

Figure 9 Federates connected to RTI through EOS (see online version for colours)**Figure 10** Banking federates exchanging data via RTI (see online version for colours)

- Step 2 (arrows numbered with '2'): a test of the final models obtained by model reverse engineering is carried out. After that, the correct models are transformed from CIM to code, and generate a federate interface, which can plug into the HLA platform and transmit the information with other processes' information systems via RTI.
- Step 3 (arrows numbered with '3'): if other federates want to join this ongoing cooperative project, they also need to follow the Step 1 and Step 2, to rewind their legacy system into MDA conceptual models, and select part of them that can be used for interoperability, then generate the federate interface, finally, synchronise with other systems.

5.4 Case study: bank's operation

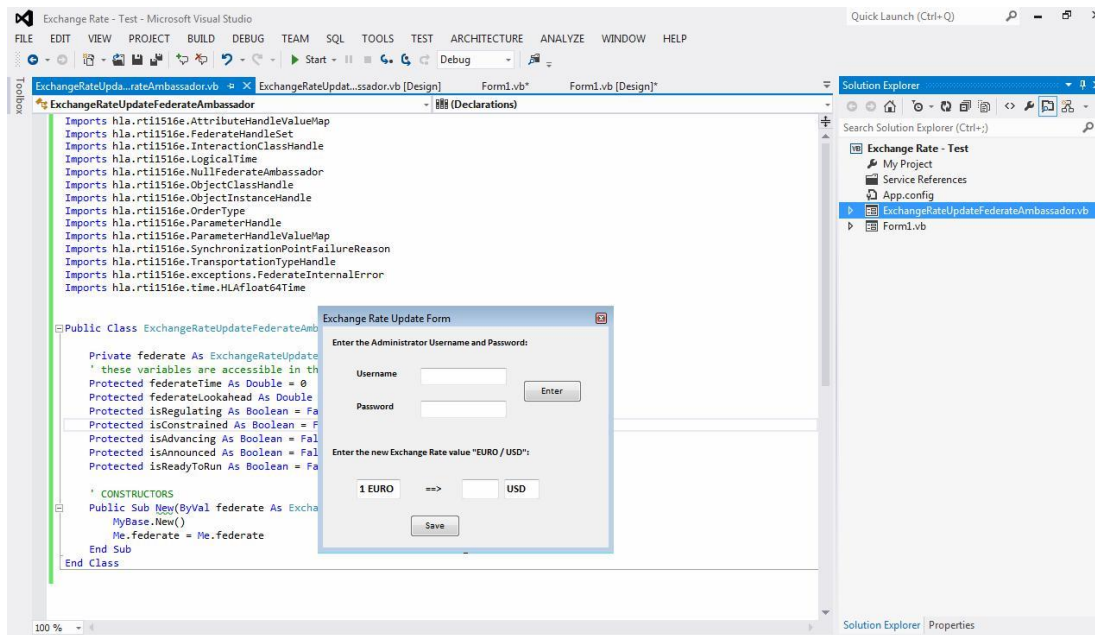
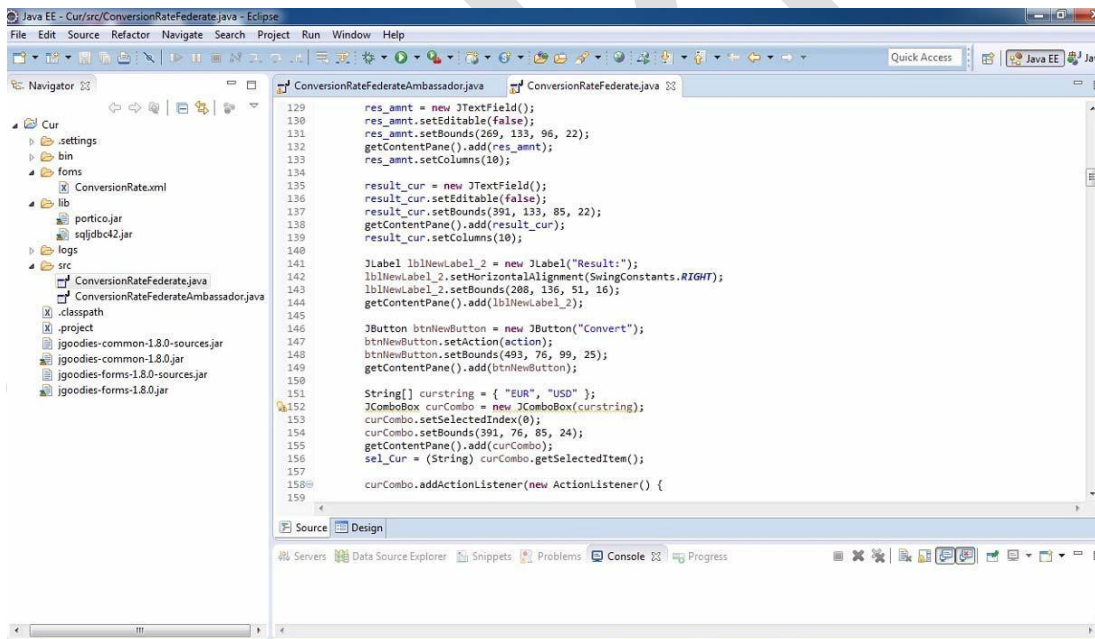
A simulation system in the banking and finance environment called 'exchange rate' has been developed and implemented as a real-world system.

The exchange rate is defined as a rate at which a country's currency is exchanged in terms of another currency. The bank's exchange rates are constantly changing once every business day based on current market conditions. In our case, the currencies exchanged are EURO and USD.

Every participating application in this system is called 'federate' that interacts with other federates through the run time infrastructure (RTI).

In our case 'poRTico' open source software was chosen as a RTI that can support heterogeneous environments. It is designed with modularity and flexibility and provides a set of software services for the dynamic information management and inheritance, in which federates coordinate their operations and exchange data during a runtime execution.

This kind of federations is provided by IEEE in order to publish and subscribe events between federates.

Figure 11 The exchange rate form federate integrated with poRTico (see online version for colours)**Figure 12** The ERM federate integrated with poRTico (see online version for colours)

As shown in Figures 8 and 9, the EOS internal components are connected together and to the external components in order to create a new activity (exchange rate update), analyse this activity, turn it as a process and execute it

through special resources and information (Youssef et al., 2016a).

Each EOS component has an essential role during the execution through the RTI in order to ensure that the

exchange rate new activity is generated based on the federated approach.

As shown in Figure 10, the ‘exchange rate form’ of the core banking system and ‘PM’ federates are fully developed and implemented using Vb.net programming language on the Visual Studio 2010 platform. The other federates, ‘EOS interface’, ‘EPM’, ‘IM’, ‘information management’, and the ‘ERM’ are fully developed and implemented using Java language on Eclipse version 6 platform.

- 1 The exchange rate form of the core banking system is connected to the related SQL storage database. It is used as a user interface that allows the bank’s branch manager to modify the daily currencies rate between EURO and USD.
- 2 The EOS interface is mainly used in order to transmit the request job, restrict any operation on the previous rate from any teller and create the process.
- 3 The EPM is mainly used in order to ensure that the login username and password are compatible with the required privileges and retrieve the exchange rate attributes.
- 4 The IM has two roles:
 - a to preserve the correct rate value from the information management federate
 - b to request from the resource management federate to assign the specified external resources (rate application, printer and teller) in order to execute the operation.
- 5 The information management is responsible of sending the new exchange rate value to the IM federate.
- 6 The ERM is mainly used in order to respond to the IM federate by checking the availability of the related resources and pre-assign it when available.
- 7 The PM is the federate that modify the exchange rate on the rate application, print out the operation receipt and notify the teller to use the new exchange rate.

The seven federates are connected together and they are communicating through poRTico in order to publish and subscribe events between modules.

Figures 11 and 12 show a sample of code related to federates that are developed in different heterogeneous environments and integrated with poRTico by calling the member functions of class RTI:RTIAmbassador which is contained in the LibRTI, and by extending and implementing the RTI:FederateAmbassador that uses pure virtual functions to send messages and requests to the RTI.

Each HLA federate ensures the execution of two parts:

- 1 The federate code block which is the source code of the federate developed, implemented and connected to the local RTI component using the .Vb and .Jar languages from the C# and the java libraries ‘LibRTI’ in order to form a complete federate.
- 2 The local RTI component code ‘LRC’ that provide the services for the federate through communication with the RTI executive component and the other federates.

As shown in Figure 13, the MoDisco software tool has been selected in our case study in order to generate a readable UML file in XML format. This tool is an open source plug-in of the Eclipse generative modelling technologies (GMT) component for model-driven reverse engineering used in order to allow practical extractions of models from legacy systems.

The UML file is generated through the ‘KDM to UML converter’ after the usage of two existing discoverers:

- 1 JavaDiscoverer which discovers KDM models from java sources
- 2 CSharpDiscoverer which discovers models from C# sources.

As shown in Figure 14, the converted UML models include packages, interfaces, classes, and also the properties and operations of classes and associations and dependencies among the classes.

In order to generate HLA FOM, a critical phase must be occurred by selecting the similar entities among the federates participating in this simulation after comparing their UML models. However, the .uml file would contain bulk information, which is not interesting for HLA FOM generation, such as external java package and common java data types. Thus, an ‘analyse UML’ sub-module is required to select the useful information, and arrange this information.

After the analysis of the UML model, the useful attributes are selected and the models are transferred into HLA FOM file by following different RTI FOM file format as shown in Figure 15. According to the poRTico RTI FOM file format, UML:Class matches to Objects:Class, and UML:Attribute matches to Objects:Attribute.

Federates interact using services proposed by the RTI. They can notably ‘publish’ to inform about an intention to send information to the federation and ‘subscribe’ to reflect some information created and updated by other federates.

This experimental case study is fully implemented in several heterogeneous environments based on the interoperability and uniformity principles in order to provide a set of domain-independent APIs used to access capabilities and features, and to exchange data between federates using the XML format.

Figure 13 MoDisco configuration and specifications (see online version for colours)

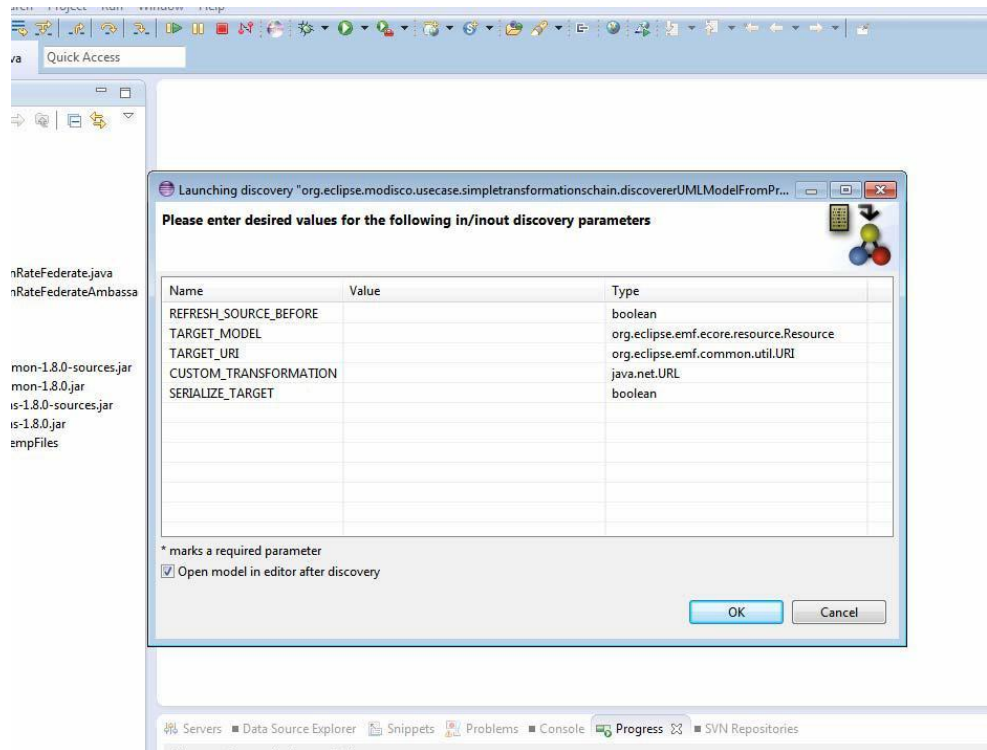


Figure 14 UML model (see online version for colours)

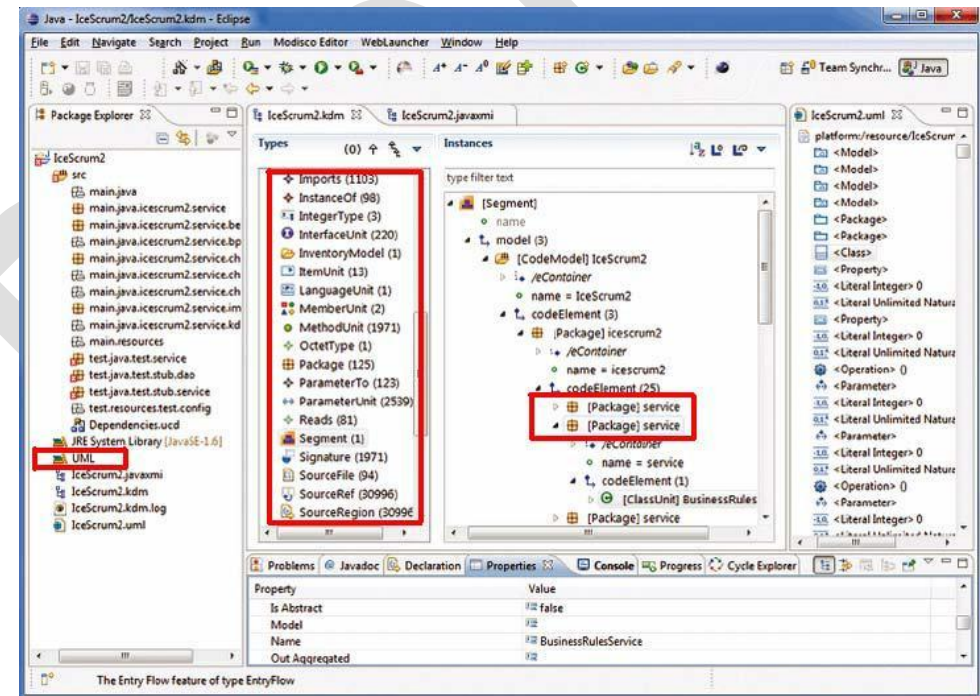
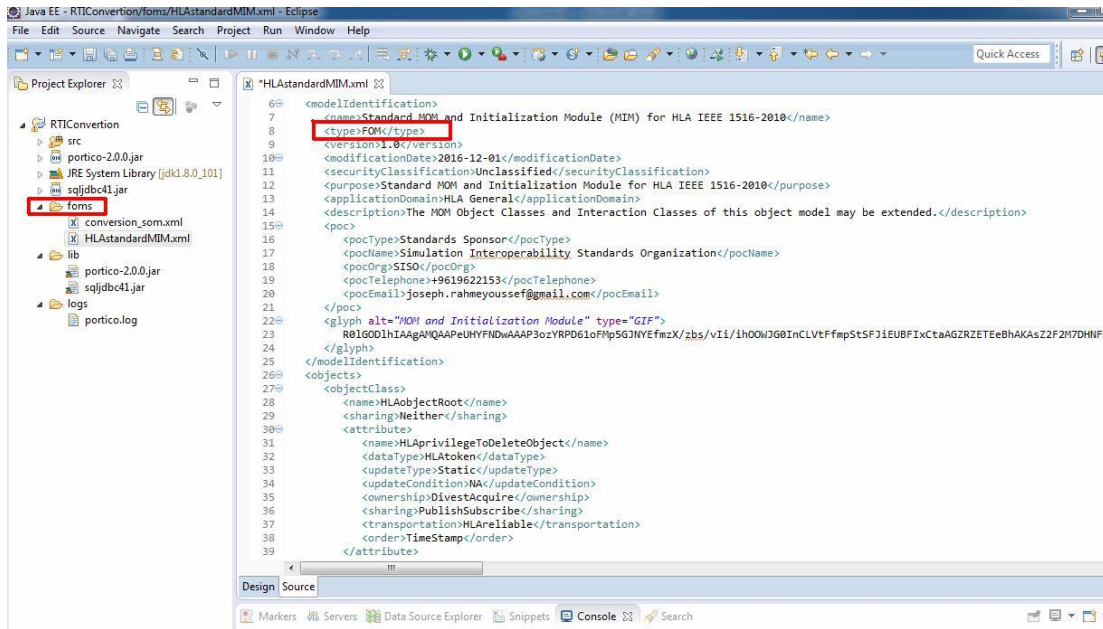


Figure 15 Generation of FOM file (see online version for colours)

6 Conclusions

This paper has tentatively presented the requirements, the conceptual, technical and development architectures to build an EOS for the new generation enterprise systems such as for example envisioned in Industry 4.0. Moreover, it also outlined the technical architecture of the reversible model driven and HLA based framework and methodology for implementing a federated approach under the enterprise interoperability framework in order to ensure the inter federates communications and the data connection between the components of the EOS and the external peripherals. This framework has the main role to support establishing enterprise interoperability dynamically in a heterogeneous and multi-partners' environment, facilitate re-use of models and re-engineering sub-systems based on models, and allowing extracting from legacy systems and software applications relevant information/data for EI engineering or re-engineering.

A case study has been designed, developed and implemented in order to test the EOS in the service sector using the federated interoperability approach for quick interoperability establishment, easy-pass, and dynamic environment update.

The proposed EOS tends to reconcile two different but complementary initiatives for enterprise management and control that exist in the market: IT platforms/infrastructure and ERP based application packages.

The reason for proposing to develop an EOS is first of all to contribute a part of the response to the foreseen enterprise evolution towards a new generation of modular

structured agile 'smart factory' based on cyber physical system principles. Traditional factory automation, empowered with internet of things (IoT), will operate a challenging industrial migration towards the Fourth Industrial Revolution (or Industry 4.0). The massive use of connected sensors and mobile devices connected to or assisting humans will make possible to collect timely real-time information throughout the enterprise. Using the real-time data on enterprise operations, events and resources, an EOS will allow the dynamic scheduling of enterprise operations with on-the-fly resources allocation for ensuring the flexibility and real-time control of enterprise functions.

The requirements presented in the paper are based on and inspired from some existing relevant approaches, in particular ENV13550 with necessary generalisation and extension to focus on the core functions of an OS for enterprise. The proposal is challenging and its success mainly depends on two factors. One is the acceptance of EOS in industry as a standard to develop an ecosystem providing varieties of enterprise application compatible to EOS; the other is the 'interoperability' service that allows other heterogeneous non-EOS compatible applications to run on EOS.

Several continuous works and tasks are being executed in order to validate and implement our proposed EOS such as:

- ☐ refining both requirements and architectures at the one hand
- ☐ developing a prototype to test the EOS against another case study in the manufacturing sector

- starting some preliminary discussion in relevant standardisation working groups such as CEN and ISO that deal with integration and interoperability.

References

- AIIM (2000) *Enterprise Applications – Adoption of E-Business and Document Technologies*, AIIM BookStore [online] http://www.techstreet.com/direct/ExSum_NA.pdf.
- Amber, S.W. (2003) 'Agile model-driven development is good enough', *IEEE Software*, Vol. 20, No. 5, pp.71–73.
- AMICE (1993) *CIMOSA Open System Architecture for CIM*, 2nd and Extended Revision, Esprit Consortium AMICE.
- Bieberstein, N., Bose, S., Fiammante, M., Jones, K. and Shah, R. (2005) *Service-oriented Architecture Compass: Business Value, and Planning*, IBM Press, ISBN: 978-0-13-187002-4.
- Bruzzzone, A., Mosca, R., Revetria, R., Bocca, E. and Briano, E. (2005) 'Agent directed HLA simulation for complex supply chain modeling', *Simulation*, Vol. 81, No. 9, pp.647–655.
- Burnson, F. (2015) *Enterprise Resource Planning Software – Buyer Report*, Software Advice™, Stamford, England.
- Buss, A. and Jackson, L. (1998) 'Distributed simulation modeling: a comparison of HLA, CORBA And RMI', *Winter Simulation Conference*.
- CEITON (2014) *Front-End and Back-End EAI*, CEITON Technologies.
- CEN (1999) *Advanced Manufacturing Technology – Systems Architecture – Enterprise Model Execution and Integration Services*, prENV 13550.
- Chang, J.F. (2005) *Business Process Management Systems, Strategy and Implementation*, CRC Press.
- Chen, D., Vallespir, B. and Doumeings, G. (1998) 'Levels of information systems interoperability', *IFAC Proceedings*, Vol. 41, No. 2, pp.12873–12878.
- Chen, D., Youssef, J.R. and Zacharewicz, G. (2015) 'Towards an enterprise operating system – requirements for standardisation', *Proceedings of IWEI 2015*.
- Elvesaeter, B., Hahn, A., Berre, A. and Neple, T. (2007) *Towards an Interoperability Framework for Model-Driven Development of Software Systems*, Interoperability of Enterprise Software and Applications.
- Fagg, G.E., London, K.S. and Dongarra, J.J. (1996) *Taskers and General Resource Managers*, Third European PVM Users' Group Meeting, EuroPVM '96, Springer, Berlin, Heidelberg.
- Favre, L. (2003) *The Workflow Management Coalition Specification, Workflow Management Coalition, Terminology & Glossary*, Baidu, Winchester, UK.
- Flurry, G. and Clark, K. (2011) *Enterprise Service Bus*, IBM – DeveloperWorks, IBM Corporation.
- Gable, J. (2002) 'Information management journal', *Magazine Article*, Questia Trusted Online Research.
- Gray, J. and Reuter, A. (1993) *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann Publishers (An Imprint Elsevier), San Francisco, California.
- Henning, M. (2006) *Response to 'The Rise and Fall of CORBA'*, ACM [online] <http://queue.acm.org/detail.cfm?id=1142044>.
- Hollingsworth, D. (1994) *The Workflow Reference Model*, Workflow Management Coalition, Document Number TC00-1003.
- IEEE (1995) *Standard for Distributed Interactive Simulation – Communication Services and Profiles*, IEEE Standards Association, IEEE Std 1278.2-1995.
- Jin, X. (2009) *Research on the Model of Enterprise Application Integration with Web Services*.
- Konstantas, D., Bourrières, J.P., Léonard, M. and Boudjlida, N. (2006) *Interoperability of Enterprise Software and Applications*, INTEROP-ESA International Conference, Geneva, Switzerland.
- Leadbeater, P.F. (1999) *Systems Architecture, Enterprise Model Execution and Integration*.
- Linthicum, D.S. (1999) *Enterprise Application Integration*, Addison-Wesley Information Technology Series, Canada.
- Massei, M., Tremori, A., Poggi, S. and Nicoletti, L. (2013) *HLA-Based Real Time Distributed Simulation of a Marine Port for Training Purposes*, DOI: 10.1504/IJSPM.2013.05 5206
- May, J.M. (2001) *Parallel I/O for High Performance Computing*, ACM Digital Library, Boston, USA.
- Moon, J. and Lee, S. (2006) 'Design and implementation of a resource management system using on-demand software streaming on distributed computing environment', *ICCS'06 Proceedings of the 6th International Conference on Computational Science – Volume Part I*, Berlin, Germany.
- Morris, E., Place, P., Plakosh, D. and Meyers, B. (2004) *System of Systems Interoperability*, Final Report, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania [online] <http://www.sei.cmu.edu/library/abstracts/reports/04tr004.cfm>.
- OMG, O. (2010) *Architecture Driven Modernization (ADM): Knowledge Discovery Meta-Model (KDM) v1.2, v1.2* [online] <http://www.omg.org/spec/KDM/1.2>.
- Omicini, A., Petta, P. and Pitt, J. (2013) 'Engineering societies', *Agents World 4th International Workshop*, London.
- Rademakers, T. and Dirksen, J. (2008) *Open-Source ESBs in Action*, Manning Publications [online] <https://www.manning.com/books/open-source-esbs-in-action>.
- Rasta, K. (2013) *Data Quality-Based Resource Management in Enterprise Service Bus*, University of Agder.
- Saha, D., Mukherjee, A. and Bandyopadhyay, S. (2011) *Networking Infrastructure for Pervasive Computing, Enabling Technologies*, Kluwer Academic Publishers; Springer Science & Business Media, Springer, Boston, USA.
- Salayandia, L. and Gates, A.Q. (2007) *Towards a Workflow Management System for Services Oriented Modules*, DOI: 10.1504/IJSPM.2007.014712
- Sethi, A.S., Raynaud, Y. and Faure-Vincent, F. (1995) 'Integrated network management IV', *Fourth International Symposium on Integrated Network Management*, Springer, US.
- Shorter, D.N. (1997) *Requirements for Model Execution and Integration Services*.
- The Open Group (2012) *SOA Reference Architecture Technical Standard: Operational Systems Layer*.
- Tolk, A. (2011) 'Bridging the data gap – recommendations for short, medium and long term solutions', *Spring Simulation Interoperability Workshop*, Orlando, Florida, US.
- Tolk, A. and Muguira, J.A. (2003) 'The levels of conceptual interoperability model', *Fall Simulation Interoperability Workshop*, U.S. Orlando, Florida.
- Turner, K.J. (2012) *Advances in Home Care Technologies, Results of the MATCH Project*, IOS Press.

- Vollmer, K. (2011) *The Forrester WaveTM: Enterprise Service Bus for Application Development & Delivery Professionals*, Forrester Research.
- Wainer, J., Filho, R.S. and Madeira, E.R.M. (2000) *CORBA Based Architecture for Large Scale Workflow*, Institute of Computing, Tokyo, Japan.
- Weatherly, R., Wilson, A. and Griffin, S. (1993) *ALSP-Theory, Experience, and Future Directions*.
- Wyszkowski, P. (2011) *ESB Application for Effective Synchronization of Large Volume Measurements Data*, Research Gate, University of Science and Technology, Kraków, Poland.
- Xu, L.D. (2014) *Enterprise Integration and Information Architecture, a Systems Perspective*, Book Depository, Toronto Public Library, Toronto, Canada.
- Youssef, J.R., Chen, D., Zacharewicz, G. and Zhiying, T. (2016a) 'Developing and enterprise operating system (EOS) with the federated interoperability approach', *I3M'16 – International Multidisciplinary Modeling & Simulation Multiconference*.
- Youssef, J.R., Zacharewicz, G. and Chen, D. (2016b) 'Developing and enterprise operating system – requirements and architectures', *ICE/IEEE'17 - International Conference on Engineering, Technology and Innovation*.
- Zacharewicz, G., Bazoun, H., Ribault, J. and Boyer, H. (2016) *SLMToolBox: Enterprise Service Process Modelling and Simulation by Coupling DEVS and Services Workflow*, DOI: 10.1504/IJSPM.2016.10001547.
- Zacharewicz, G., Chen, D. and Vallespir, B. (2009) *Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability*.
- Zaraté, P. (2013) *Tools for Collaborative Decision-Making*, Wiley Online Library [online] <http://onlinelibrary.wiley.com/doi/10.1002/9781118574690.ch2/summary>.
- Zhiying, T., Chen, D. and Zacharewicz, G. (2012) *Federated Approach for Enterprise Interoperability: A Reversible Model Driven and HLA Based Methodology*.
- Zurawski, R. (2004) *The Industrial Information Technology Handbook*, Handbook. CRC Press Book.