DEVELOPING AN ENTERPRISE OPERATING SYSTEM (EOS) WITH THE FEDERATED INTEROPERABILITY APPROACH

Joseph Rahme Youssef[(a)], Gregory Zacharewicz[(b)], David Chen[(c)], Zhiying Tu[(d)]
[(a)(b)(c)]University of Bordeaux, 351, Cours de la libération, 33405 Talence cedex, France
[(d)]School of Software, Harbin Institute of Technology, 92 West Dazhi Street, Nan Gang District, Harbin, China

[(a)]joseph.youssef@ims-bordeaux.fr,[(b)]gregory.zacharewicz@ims-bordeaux.fr,[(c)]david.chen@ims-bordeaux.fr,
[(d)]tzy_hit@hit.edu.cn

ABSTRACT
Furthermost of enterprises have chosen to implement ERP solutions or multiple systems in order to facilitate the data orchestration by connecting several software and hardware together at the operational level [1]; Nevertheless this solution may constraint the business due to the top down "enclosing" methodology. Another approach may take place by setting loose coupled connections between enterprise's software in the idea of federated interoperability with only one simplified central orchestrator component. The cooperating parties must accommodate and adjust "on-the-fly" to ensure quick interoperability establishment, easy-pass, and dynamic environment update. In that objective, developing an Enterprise Operating System (EOS) is seen as one of the key steps towards the future generation enterprise manufacturing systems based on IoT and Cyber Physical System principle. This paper tentatively presents at first a set of requirements and functionalities of EOS. Then a survey on existing relevant works is presented 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, Architecture, Model, Infrastructure, Interoperability.

1. INTRODUCTION
This paper aims at presenting an ongoing research activity carried out at IMS laboratory of University of Bordeaux to develop an Enterprise Operating System (EOS) based on the federated enterprise interoperability concept in order to strengthen the performance and competitively of small and medium manufacturing enterprises and improving the effectiveness of collaborative environments. It presents the requirements and architecture of an Enterprise Operating System (EOS) and the technical architecture of the Federated Interoperability framework.

The proposed EOS will behave in the same way as an Operating System of the computer, but under enterprise context. This EOS will act as a system-wide interface between enterprise business managers and enterprise operational resources performing daily enterprise operations.

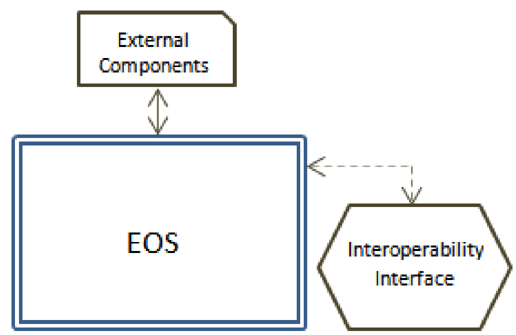

Figure 1. The Enterprise Operating System, Interoperability Interface and the external peripherals

To make this highly flexible, dynamically configured and real time controlled enterprise a reality, a sound Enterprise Operating System (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 massive use of sensors, actioners and robots coming from different vendors will also require an EOS that is capable of connecting them together to work as a 'one'.

Today, enterprise operational management is largely dominated by integrated solutions like ERP. It has been estimated that 78% of enterprises have chosen ERP solutions or multiple systems in order to facilitate the data orchestration by connecting several software and hardware together at the operational level [1]. Nevertheless this solution may constraint the business due to the top-down "enclosing" methodology.

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). It will interpret content of enterprise models defined by business managers, trigger various enterprise operations with dynamically allocated enterprise resources, and monitor the status of enterprise resources (Human, Machining, Computing) through various sensing devices and front-ends [2][3].

In other words, EOS and ERP have similar objectives but different approaches. ERP is a top-down integrated approach incorporating in its framework all enterprise applications and needed enterprise operating functions (at least they intend to do this); While EOS is a bottom-up federated approach only providing enterprise operating system (functions) that allows various heterogeneous applications from different vendors to connect to EOS and to work together.

From what have been stated above, we believe that EOS ‘s approach presents the better perspective for the future in the context of Internet of Things (IoT) and Factory of Future (FoF). It also provides an alternative for those enterprises with a tailored solution, especially for SMEs (Small and Medium-sized Enterprises). A SME equipped with an EOS (cost much less than ERP) will be able to only purchase the applications they need (possibly from different vendors). Once EOS is adopted in the industry with the Federated Interoperability approach, a foreseen ecosystem would be developed in consequence to provide varieties of enterprise applications compatible to EOS, just like what happened with Apple’s ecosystem to IOS (iPhone Operating System) and Google’s one to Android. To make this happen, the envisioned EOS needs to be recognized as a standard so that enterprise solutions vendors develop their applications according to the specifications of the EOS.

2. REQUIREMENTS AND FUNCTIONALITIES OF EOS

This section presents an outline of the requirements and functionalities for developing an Enterprise Operating System.

2.1 Enterprise Resource Management “ERM”

Enterprise Resource Management is essential for EOS in order to monitor enterprise resources system-wide (available, occupied, out-of-order…) [4]. It provides a real-time and global view of the use of those resources in a company. Main required functionalities of ERM are: checking, searching, reporting and selecting the availability of resources; matching the required capabilities to the capabilities of existing available resources; allocating and de-allocating resources; and ensuring that the right resources are allocated to the right place at the right time.

2.2 Enterprise Process Management “EPM”

Enterprise Process Management is required for EOS to execute enterprise processes defined by the business managers and other EOS internal process in order to carry out enterprise operations. Main functionalities of EPM include: sending commands triggering the starting of processes, recording ending status of processes (done, not done, fail…); monitoring the activity and state of individual enterprise processes and key performance indicators; and interacting with the Resource Management in order to send commands, interpret and receive resources statuses [4].

2.3 Enterprise Information Management “EIM”

Enterprise Information Management supports information and data exchange between all entities connected to the EOS such enterprise resources and business managers [4].

Main required functionalities of EIM are: ensuring the centralized management of reference data used; excluding data duplicates; ensuring and maintaining system-wide consistency as well as integrity of enterprise data exchanged; providing transparent access to data sources and an appropriate data storage facility needed for running EOS; ensuring information and data confidentiality and security to protect from non-authorized access.

2.4 Presentation Management “PM”

Presentation Management is concerned with the interface between internal and external worlds of EOS. It is mainly required to organize and coordinate the communication and information flow between Enterprise Resources and internal entities of EOS including the Human, Machine and Applications Dialogue services [5].

2.5 Interoperability Management “IM”

Interoperability Management is a utility service of EOS. Interoperability is a precondition for a successful development and implementation of the Enterprise Operating System by ensuring that EOS can interoperate with both Business Software Applications and the Embedded Software provided for device controllers and sensors. The federated approach may be helpful for EOS by allowing quick interoperability establishment, easy-pass, and dynamic environment update in the heterogeneous and multi-partners environment.

3. EXISTING WORKS RELEVANT TO EOS

Since the beginning of 80’s, a lot of works has been done to develop IT infrastructures and platforms to support enterprise activities. Some of them are relevant to EOS.

3.1 Existing Works

CORBA (Common Object Request Broker Architecture) was designed to facilitate communication of systems deploying on divers- platforms, written in different languages and executed on different operating systems. CORBA is seen more as an enterprise application integration platform rather than an Enterprise Operating System [5]. Compared to CORBA, ENV 13550 moved a step towards an EOS. ENV 13550 is a pre-European standard based on CIMOSA enterprise modeling and integration concepts. It aimed at supporting business process monitoring and control [4]. ENV 13550 focused on Enterprise model execution; integration of divers’ heterogeneous applications is not well supported as CORBA [7]. Another well-known work similar to CORBA is EAI

(Enterprise Application Integration) that is an integration framework defined by Gartner Group. It is composed of a collection of technologies and services to provide integration of systems and applications that reside on different operating systems and use different database solutions across an enterprise (SCM, ERP, CRM, payroll…). As CORBA, EAI was not dedicated to enterprise process and operation monitoring and control that are to be supported by EOS [8]. Concerning this last point, we must mention the work of WfMC (Workflow Management Coalition) to define standards for the interoperability of Workflow Management Systems [9]. Similar to 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 operating systems to distribute processing and data across the enterprise by providing a coherent environment for developing and running applications [10]. ODP is an international standard published in 1990 to describe and build widely distributed systems and applications in a multi-vendor environment [11]. 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 ESB (Enterprise Service Bus) and SOA (Service Oriented Architecture). ESB is a software architecture model defined in 2002 as a set of rules and principles to enable interoperability between heterogeneous environments [12]. While SOA is an architectural style defined by the Open Group to support service orientation including service-based development and outcomes services. Both ESB and SOA cannot directly be used as an EOS but they provide interesting concepts and principle to develop and build the EOS [13].

3.2 Efficiency of existing Techniques for EOS

Considering Enterprise Resource Management, most of the existing approaches only focus on IT resource management (monitoring and control). Only ENV 13550 and to a lesser extent the WfMC deal with other non IT enterprise resources such as 'Human' and 'Machine'. SOA allows discovering and matching required capability to existing available resources; this is a useful approach for EOS [14, 15, 16, 17, 18, 19].

Referring to Enterprise Process Management, most of the existing approaches include Process Management. However, distinction should be made between 'Business Process' defined by business users and 'Process' that is orchestrated to ensure internal working of platforms. In an EOS, both the two types of processes will exist 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 [20, 21, 22, 23, 24, 25].

For Enterprise Information Management, information exchange between all entities connected to an EOS is essential function. All existing approaches support this function. However due to the heterogeneous data sources in an enterprise, interoperability is fundamental to support this function [26, 27].

Concerning the Presentation Management being not a core function for an EOS but necessary to interact with all resources and business users connected to an EOS, ENV 13550 presents the most interesting concepts, as it allows to dialogue with all types of enterprise resources (human, machine and IT) [28, 29, 30].

For Interoperability Management, existing approaches focused on Enterprise Integration rather than loosely coupled interoperability. In most existing approaches interoperability is not developed to a mature and satisfactory level. A set of interoperability utility services needs to be developed to support the use of the EOS [31, 32, 33, 34, 35].

In conclusion, the existing works studied are all relevant to developing an EOS but they don't cover all the requirements and functionalities. Each of them has its own objectives with specific focuses. The results of the comparison have been quantified and summarized in the figure 2.

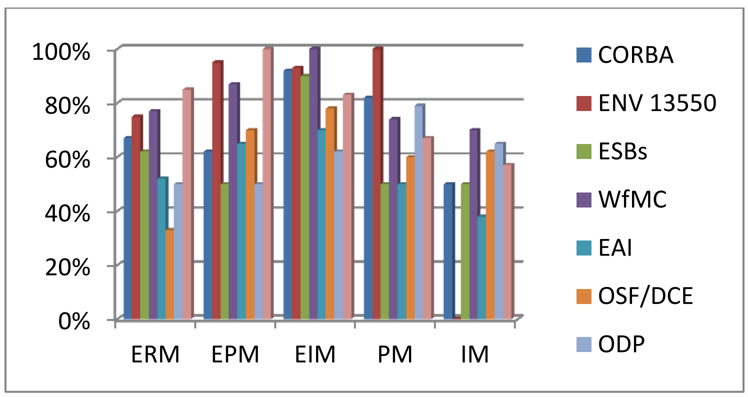

Figure 2. Summary of the existing Techniques' services and efficiencies based on the EOS requirements

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

LISI focuses on technical interoperability and the complexity of interoperations between systems. But LISI model does not address the environmental and organizational issues that contribute to the construction and maintenance of interoperable systems. 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 organizational layer [36]. The Database interoperability & 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 [37]. The LCIM model has been carried out successfully in simulation domain, but the basic premises apply to many complex sets of interoperating systems [38]. The SOSI model extends the existing models by adding a focus on programmatic, constructive and operational issues which must be managed across the life cycle [39].

The MDA approach contributes on building an interoperable ICT model, from enterprise models to technology models. Those models are able to be aligned by using common meta-model. 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 percent effort to generate incomplete and useless code (80 to 90 percent), but spend 90 percent 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 [40]. The soundness of the 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 [41]. 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. ADM met the same model transformation problems as MDA [42].

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 [43][44][45][46].

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 [47].

4.2 Efficiency of Models for Federated Interoperability Concept

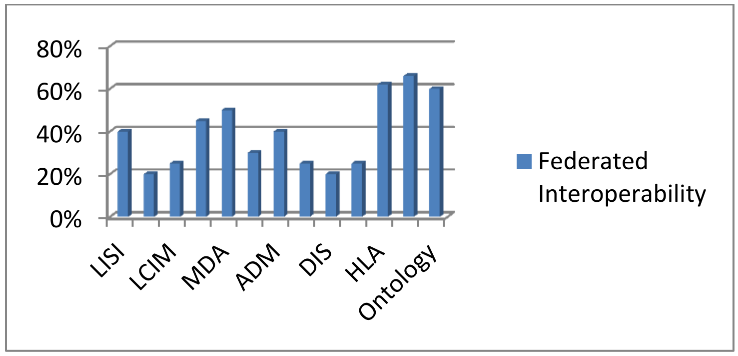

Figure 3. Summary of the Models' services and efficiencies based on the Federated Interoperability concept

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 harmonized and reversible HLA based methodology is being implemented for developing 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.

5. ARCHITECTURES

This section outlines the conceptual architecture of EOS, technical architecture of EOS and the technical architecture of the Harmonized/reversible development framework in order to meet the requirements and functionalities identified in section 2 and 3.

5.1 EOS Conceptual Architecture

Figure 4 describes the EOS Conceptual Architecture. This EOS is a distinctive system-wide platform that allows the business managers to communicate and operate through the systems' (hardware, software, network, machines…) in an efficient and effective way [2].

Unlike ERP, the Enterprise Operating System EOS will mainly be developed and implemented as one simplified central orchestrator component connected to several peripheral devices and external components.

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. They are commercial and purchasing agents, product designers, production managers, shop floor operators etc...

IT type resources 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, inventory management software etc…
Machine type resources are material transformation and processing devices and equipment 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 enables business interoperability between EOS executed systems and the external components related to the EOS.

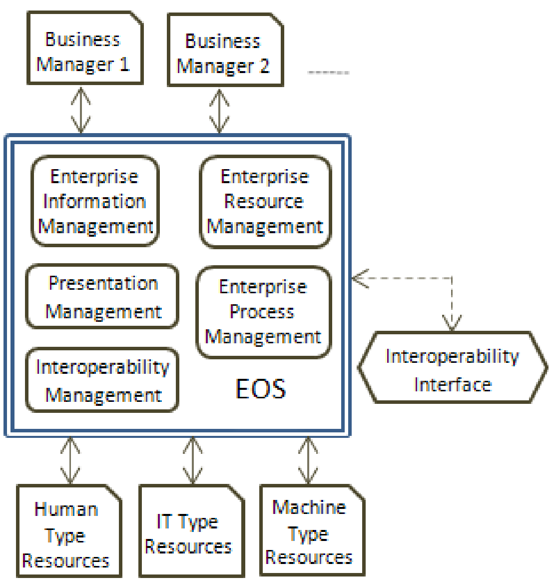

Figure 4. EOS Conceptual Architecture

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.
Enterprise resource management dynamically monitors the status of enterprise resources, search and allocate suitable resources to operations that must be done.
Enterprise process management executes business processes defined by business users, coordinates and executes EOS internal processes/operations.
Enterprise information management manages, protects and supports information and data exchange of all kinds between the enterprise's resources connected to the EOS.
Presentation management is a set of services with appropriate interfaces that allow business users and other enterprise resources to connect to EOS and receive/send information.

Interoperability management is a set of services that provide necessary mapping between heterogeneous resources to make them interoperable through EOS.

5.2 EOS Technical Architecture

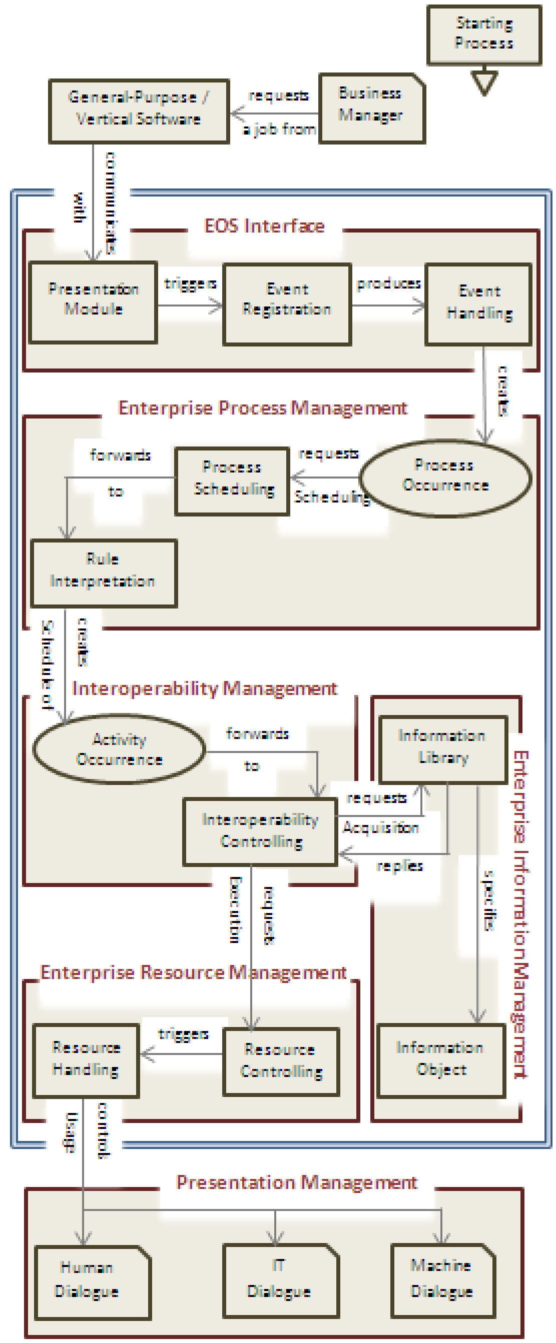

Figure 5. EOS Technical Architecture

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 send special commands to communicate with the EOS Front-End interface called Presentation Module in order to execute the requested job.

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.

Next, the Event Handling - manages the events' priorities, queues and traceability, -provides the Order Identifier, and creates the Process Occurrence. The Process Occurrence requests scheduling from the Process Scheduling component which interprets the Process Behavioral, Information and Resource requirements. This sub-service checks the authorization to execute the Process, retrieves Process descriptions from the Run-Time Repository, invokes the Enterprise Resource Management 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 initialize 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 Enterprise Resource Management to assign resources allocated by the Process Scheduling Component, invoking the Enterprise Information Management to acquire the Object States and specify the Information Object required, requesting from the Enterprise Resource Management to release the involved Resources when terminating an activity, and signaling 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 select the appropriate resource after matching the capabilities required and by taking into consideration the time, performance and priority.

The Presentation Management 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 authorized 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 Case Study: Bank's Operation

A simulation system in the Banking and Finance environment is presented, validated and being progressively implemented as a real-world system. The exchange rate is defined as a rate at which a country's currency will be exchanged in terms of another currency. A Bank's exchange rates are constantly changing once every business day based on current market conditions.

Figure 6 and 7 describe the Conceptual and Technical Architectures of the Bank's daily Exchange Rate update.

As shown in figure 7, the first federate is the Core Banking System which is developed using Java language and SQL based database. The second one is the EOS Interface Java based platform which plays the role of the presentation module of the EOS. The third federate is the Enterprise Process Management presented as a Java module. The three federates are connected together through a HLA Federation in order to update the Exchange Rate in the "Start of Day" stage.

Each one is composed of two blocks: 1) The Code Block which contains the Federate Ambassador that uses pure virtual functions to send messages and requests to the RTI (Run Time Infrastructure). 2) The Local RTI Ambassador provides the services for the related federate through communication with poRTIco RTI application which play the role of the RTI component in this HLA Federation.

poRTIco RTI component is the fundamental component used to implement the High Level Architecture in order to coordinate federates' operations and exchange data. This middleware contains a Central RTI component "CRC" connected to the Local RTI component "LRC" of each federate in order to convert requests into messages to be sent between federates. It supports HLA simulation development to greatly expand the use of distributed simulation between the Core Banking System, the EOS Interface, and the Enterprise Process Management.

The Core Banking System sends the Exchange Rate job to the Run Time Infrastructure through the Core System Code Block. The LRC1 transmits the job to the EOS Interface by notifying the LRC2. The EOS Interface creates the process of Exchange Rate's update and sends it to RTI through the EOS Interface Code Block. The RTI notify the LRC3 about the new event and then the Bank's Process Management checks the privileges of the Bank's Manager, retrieves the Exchange Rate's information and sends it to the RTI.

The Central RTI Component manages the federation by communicating with the LRC of each federate in order to update, reflect, send and receive data between federates.

This Technical Architecture is fully implemented in the Java language 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.

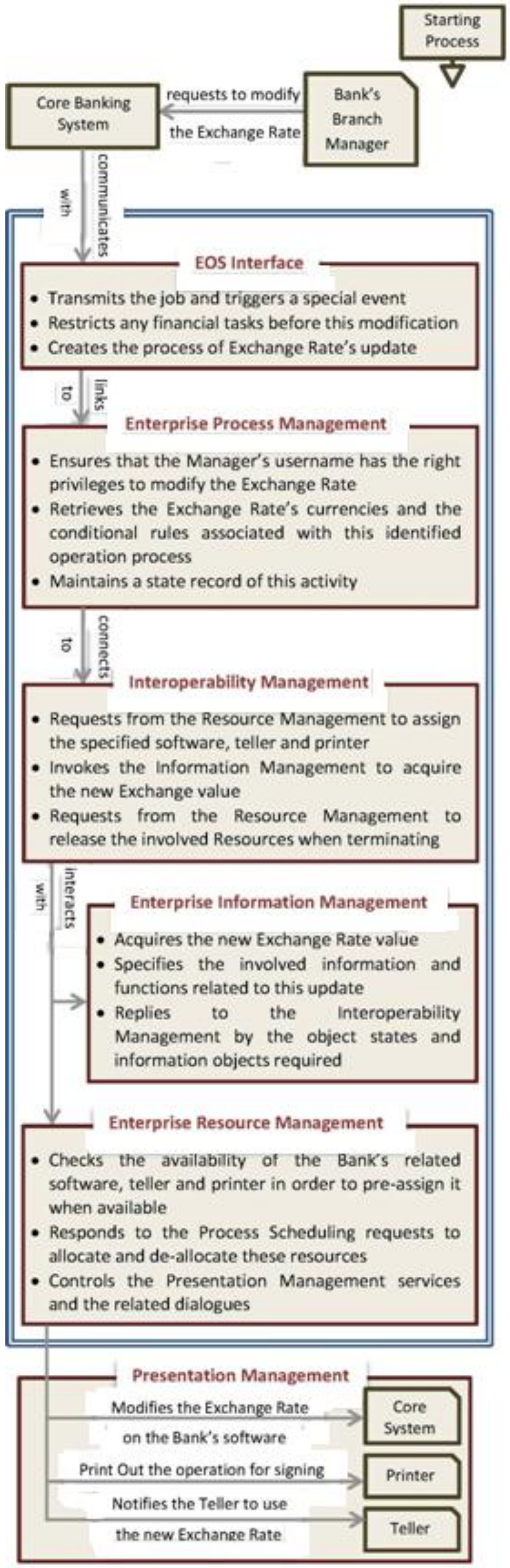

Figure 6. Conceptual Architecture of the Bank's Exchange Rate update

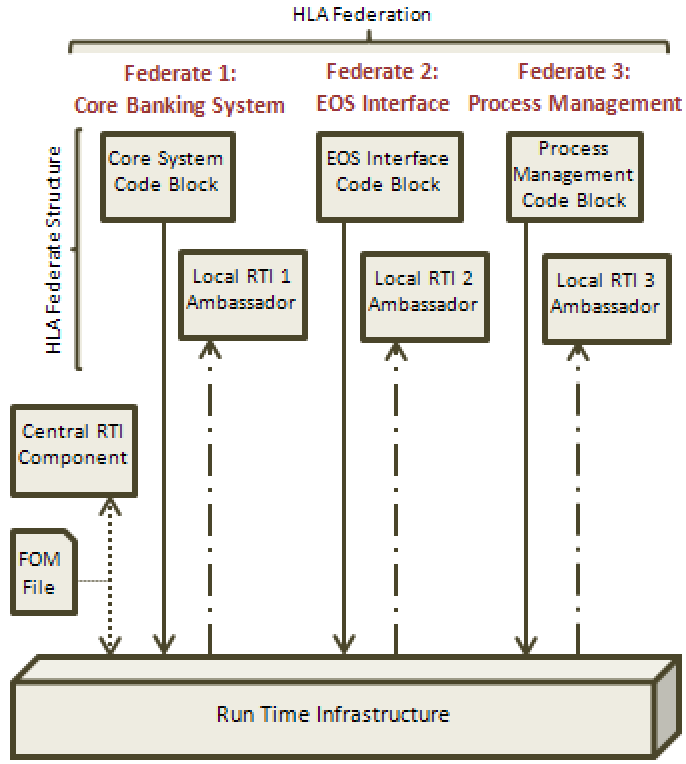

Figure 7. Technical Architecture of the Bank's Exchange Rate update

5.4 Harmonized and Reversible Development Framework – Interoperability Interface for EOS

A new framework is presented and will be used as an Interoperability Interface connected to the Enterprise Operating System 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 utilize 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.

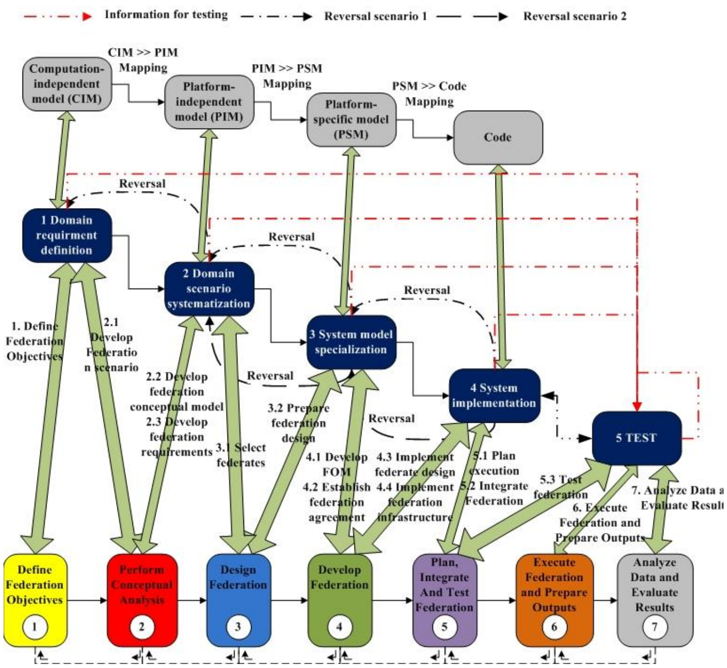

Figure 8. Harmonized and reversible development framework for HLA based Application

Primary concepts are separately presented as follow:

- Harmonized: means this framework is a synthetic framework, which consists of several techniques. As the framework in figure 8 shows, we propose a new five steps development life cycle 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 [41]. 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. The Web Services allows 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 8 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. Section 3.3 will present the method of using model reverse technique

- 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 [48]. 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 Harmonized part, Web Services will be used to improve the limitation of the traditional HLA.

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:

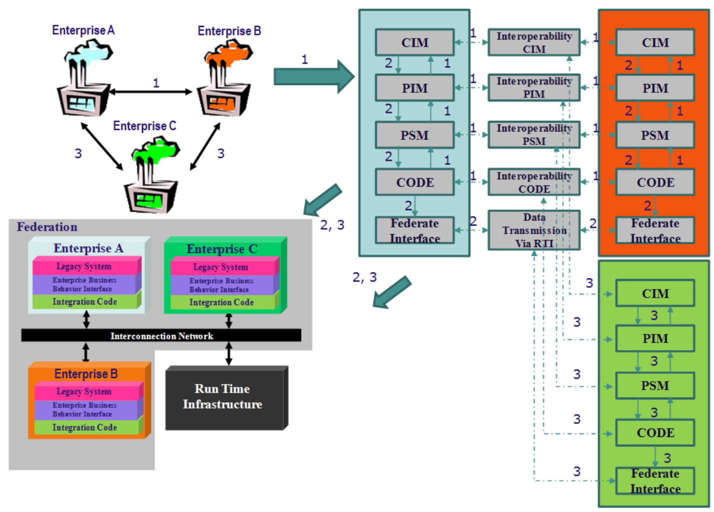

Figure 9. Scenario Description

- 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.

- 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, synchronize with other systems.

6. CONCLUSION

This paper has tentatively presented the requirements and architecture to develop an Enterprise Operating System (EOS) for the new generation enterprise systems such as for example envisioned in Industry 4.0. On the other hand, it presents the technical architecture of the Reversible Model driven and HLA based framework and methodology for implementing federated approach under the Enterprise Interoperability Framework used 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.

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

The requirements presented in the paper is based on and inspired from some existing relevant approaches, in particular ENV13550 with necessary generalization and extension to focus on the core functions of an Operation System 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 one is the 'Interoperability' service that allows other heterogeneous non-EOS compatible applications to run on EOS.

Future work planned is to refine both requirements and architectures at the one hand, and on the other hand to develop a prototype to test the EOS against two use cases in both manufacturing and service sectors using the Federated Interoperability Approach for allowing quick interoperability establishment, easy-pass, and dynamic environment update.

REFERENCES

[1] Burnson F. (2015), Enterprise Resource Planning Software - Buyer Report

[2] Chen, D., Youssef, J.R. and Zacharewicz, G. (2015), Towards an Enterprise Operating System – Requirements for Standardisation, Proceedings of IWEI 2015.

[3] Zhiying, T., Chen, D., and Zacharewicz, G. (2012), Federated Approach for Enterprise Interoperability: A Reversible Model driven and HLA based methodology

[4] Leadbeater, P.F. (1999), Systems Architecture, Enterprise Model Execution and Integration, CEN/TC310, ENV13550.

[5] Gray, J. and Reuter, A. (1993), Transaction Processing: Concepts and Techniques.

[6] Bryan, K., Dipippo, L.C. and Fay-Wolfe, V. (2005), Integrated CORBA Scheduling and Resource Management for distributed Real-Time Embedded Systems.

[7] Shorterl, D.N., (1997), Requirements for Model Execution and Integration Services, CEN/TC310 WG1.

[8] Linthicum, D.S. (1999), Enterprise Application Integration.

[9] Hollinsworth, D. (1994), The Workflow Reference Model.

[10] Turner, K.J. (2012), Advances in Home Care Technologies, Results of the MATCH Project.

[11] Zurawski, R. (2004), The Industrial Information Technology Handbook

[12] Flurry, G. and Clark, K. (2011), Enterprise Service Bus.

[13] The Open Group, (2012), SOA Reference Architecture Technical Standard: Operational Systems Layer.

[14] Zaraté, P. (2013), Tools for Collaborative Decision-Making.

[15] Rademakers, T and Dirksen, J. (2008), "Open-Source ESBs in Action".

[16] Favre, L. (2003), The Workflow Management Coalition Specification, Workflow Management Coalition, Terminology & Glossary.

[17] AIIM, (2000), Enterprise Applications - Adoption of E-Business and Document Technologies.

[18] Moon, J. and Lee, S. (2006), Design and Implementation of a Resource Management System Using On-Demand Software Streaming on Distributed Computing Environment.

[19] Bieberstein, N. (2005), Service-oriented Architecture Compass: Business Value, and Planning.

[20] Henning, M. (2006), Response to 'The Rise and Fall of CORBA'.

[21] Vollmer, K. (2011), The Forrester Wave™: Enterprise Service Bus for Application Development & Delivery Professionals.

[22] Sethi, A.S., Raynaud, Y. and Faure-Vincent F. (1995), Integrated Network Management IV, Proceedings of the fourth international.

[23] Gable, J. (2002), Enterprise application integration, Information Management Journal.

[24] Fagg, G.E., London, K.S. and Dongarra, J.J. (1996), Taskers and General Resource Managers.

[25] Chang, J.F. (2005), Business Process Management Systems, Strategy and Implementation.

[26] Wainer, J., Filho, R.S. and Madeira, E.R.M. (2000), CORBA Based Architecture for Large Scale Workflow.

[27] Rasta, K. (2013), Data Quality-Based Resource Management in Enterprise Service Bus.
[28] Wyszkowski, P. (2011), ESB application for effective synchronization of large volume measurements data.
[29] CEITON, (2014), "Front-end and back-end EAI", CEITON technologies.
[30] Jin, X. (2009), Research on the model of enterprise application integration with web services.
[31] Omicini, A., Petta, P. and Pitt, J. (2003), Engineering Societies in the Agents World IV, 4th International.
[32] Konstantas, D., Bourrières, J.P., Léonard, M. and Boudjlida, N. (2006), Interoperability of Enterprise Software and Applications.
[33] Saha, D., Mukherjee, A. and Bandyopadhyay, S. (2011), Networking Infrastructure for Pervasive Computing, Enabling Technologies.
[34] May, J.M. (2001), Parallel I/O for High Performance Computing.
[35] Xu, L.D. (2002), Enterprise Integration and Information Architecture, A Systems Perspective.
[36] Chen, D., Vallespir, B., Doumeingts, G. (1998), Levels of Information Systems Interoperability.
[37] Tolk, A. (2001), Bridging the Data Gap - Recommendations for Short, Medium and Long Term Solutions.
[38] Tolk, A. (2003), The Levels of Conceptual Interoperability Model.
[39] Morris, E., Place, P., Plakosh, D., Meyers, B. (2004), System of Systems Interoperability.
[40] Amber, S.W. (2003), Agile model-driven development is good enough.
[41] Elvesaeter, B., Hahn, A., Berre, A., Neple, T. (2007), Towards an Interoperability Framework for Model-Driven Development of Software Systems.
[42] OMG (2010), Architecture Driven Modernization (ADM).
[43] Buss, A., Jackson, L. (1998), Distributed Simulation Modeling: A Comparison Of HLA, CORBA And RMI.
[44] IEEE (1995), Standard for Distributed Interactive Simulation - Communication Services and Profiles.
[45] Weatherly, R., Wilson, A., Griffin, S. (1993), ALSP-theory, experience, and future Directions.
[46] Bruzzone, A.G., Longo, F. (2013), 3D simulation as training tool in container terminals: The TRAINPORTS simulator. Journal of Manufacturing Systems, Volume 32, Issue 1, January 2013, Pages X85-98
[47] Zacharewicz, G., Chen, D., Vallespir, B. (2009), Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability.
[48] poRTIco RTI - www.porticoproject.org