



Enterprise Operating System (EOS) in Action: Distributed Simulation of Enterprise Activities and Operations

Joseph Youssef, Grégory Zacharewicz

► To cite this version:

Joseph Youssef, Grégory Zacharewicz. Enterprise Operating System (EOS) in Action: Distributed Simulation of Enterprise Activities and Operations. WinterSim 2019 - Winter Simulation Conference, INFORMS, Dec 2019, National Harbor, United States. pp.90-104, 10.1109/WSC40007.2019.9004958 . hal-02403474

HAL Id: hal-02403474

<https://hal.science/hal-02403474>

Submitted on 9 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ENTERPRISE OPERATING SYSTEM (EOS) IN ACTION: DISTRIBUTED SIMULATION OF ENTERPRISE ACTIVITIES AND OPERATIONS

Joseph Youssef

American University of Beirut
Riad El Solh, 1107-2020
Beirut, LEBANON

Gregory Zacharewicz

IMT – Mines Alès,
6 Avenue de Clavières,
Alès, FRANCE

ABSTRACT

Operating System (OS) is a well-known concept in computer science acting as an interface between human and computer hardware. 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).

Unlike ERP, which is defined as a platform that allows the organization at the operational level to use a system of integrated applications in order to automate many back office functions related to technology and services; and unlike the Best of Breed (BOB), which is defined as a leading software that focuses on providing the features and functions for one component in the value chain by maintaining multiple systems; EOS will act as an interface between enterprise business managers and resources for supporting the distributed simulations and allowing interoperability over the heterogeneous environments by ensuring real time monitoring and control of enterprise operations.

1 INTRODUCTION

Nowadays, daily enterprise operations are not effectively monitored and controlled. This is mainly due to the lack of a system-wide operating system, the variety of heterogeneous enterprise applications and the difficulties to obtain real-time data of enterprise operations.

Industrial organizations operating in the market are becoming increasingly complex and dynamic due to the arrangement, coordination, and management of complex computer systems, middleware and services. These enterprises are constantly redesigning their structures and business processes to practice enterprise application integration in order to accommodate and implement their increasing services, data orchestration and distribution methods. On the other hand, rapid changing market requirements will oblige an enterprise to be small, smart and scalable to be able to operate quick change.

With the development of the next generation enterprise manufacturing systems based on Internet of Things (IoT) and Cyber Physical System (CPS) principle, future enterprises will need tailored “plug & play” interoperable solutions to quickly configure their systems and to real-timely monitor and control their daily operations.

There are many initiatives to develop enterprise integration platforms and infrastructures such as CORBA, EAI, etc. However, those initiatives fail to provide satisfactory solutions to integrate heterogeneous enterprise application software, especially for SMEs. Enterprise integration is still seen as a complex, costly and time-consuming task.

On the other hand, packaged-software vendors, provided by ERP vendors and other products (PeopleSoft, SAP and JD Edwards), concentrated on one application or a suite of applications that they automated without regard to other possible applications that a company might have in its systems portfolio (Burnson, 2015). Consequently, large corporations that purchased and installed packaged applications (such as ERP packages) found themselves with islands of data and processing that must be bridged to other islands. As these bridges grew, the maintenance of a system and its interfaces also grew to take up more and more resources. Facing the problems in enterprise integration and ERP packaged solution; a different

approach consists in developing an Enterprise Operating System (EOS) that aims at joining existing approaches together for the sake of complementarity and synergy. It will behave by setting loosely coupled connections between enterprise's software in the idea of federated interoperability with only one simplified central orchestrator component. It is also seen as one of the key steps towards the future generation enterprise manufacturing systems to provide an alternative for enterprises with a tailored solution especially for Small and Medium-sized Enterprises.

EOS will execute enterprise models defined by business managers, trigger enterprise operations with dynamically allocated enterprise resources and monitor the use of enterprise resources (Human, Machining, Computing) through various sensing devices and front-ends (Chen, Youssef, & Zacharewicz, 2015).

HLA software architecture is a supportive framework for the distributed simulations and will be implemented for "EOS" in order to rapidly develop HLA based interface for achieving federated enterprise interoperability, specify the object model used by the federation, provide facilities for allowing federates to interact with each other and managing the execution (Federation management, Time management, Data Distribution management,...), enable interoperability among different operating systems, and allow events to interact in an EOS (Muller, 2006; Zacharewicz, Chen, & Vallespir, 2008, 2009).

This tutorial tentatively presents at first a comparison between EI, ERP, BOB and EOS concepts. Then the position of EOS based on Interoperability is presented and described. After that the set of requirements and functionalities of EOS are defined. The architectures of envisioned EOS are outlined. Next a case study in the banking sector is described in order to show the feasibility of the EOS and illustrate the functioning of the technical and implementation architectures elaborated. The last part draws some conclusions and presents the benefits and challenges of EOS.

2 COMPARISON BETWEEN EI, ERP AND EOS

The Enterprise Integration (EI) has been developed and implemented in the 1980's to be able to rapidly respond to the changing environments, the progress in the information technology, and the growth and complexity of the enterprise operations (Gable, 2002; Linthicum, 1999). However, the integration of different systems required rewriting codes on source and target systems, consuming much time and money. Moreover, this concept failed to ensure the full integration and intercommunication and especially between enterprise's heterogeneous applications and their related operating systems (as illustrated in figure 1-A).

A good number of enterprises have chosen ERP solution in order to facilitate the data orchestration and especially in the 1990's and 2000's (Diercksen, 2012); however, this platform presents several major insufficiencies in providing the federated interoperability, and because of becoming enormously in size and complexity which make the implementation and the updates very time consuming, involving high costs and disrupt current processes and operations.

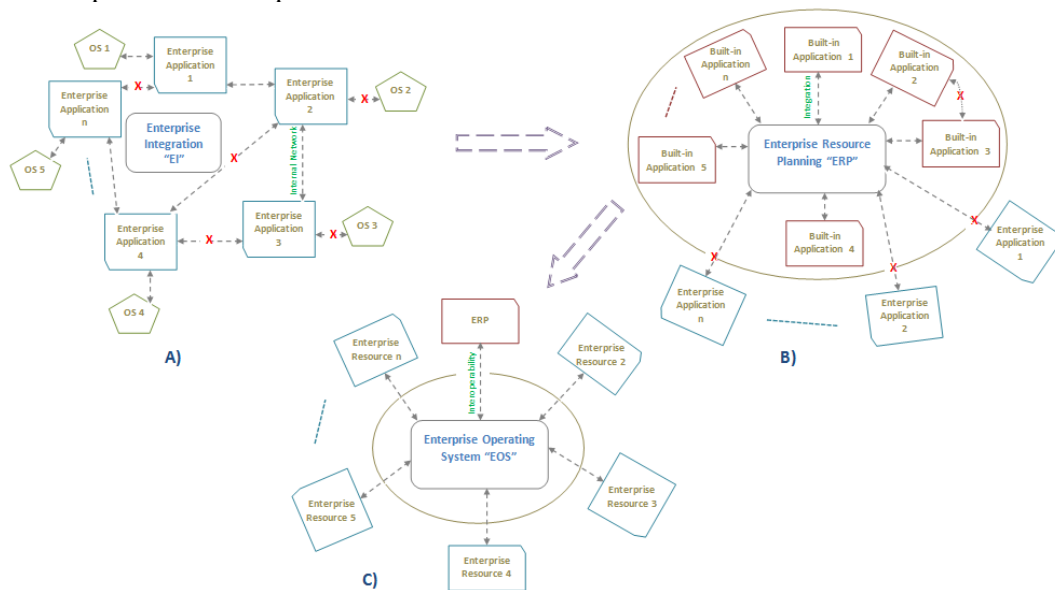


Figure 1: Comparison between EI, ERP and the proposed EOS.

As shown in figure 1-B, the ERP module is basically presented as a black-box application implicating built-in services and modules used in order to manage and control the enterprise's operations and activities; however, ERP does not ensure an efficient integration between its built-in applications. Moreover, this platform cannot ensure the compatibility and effectiveness of the direct intercommunication between enterprise resources.

On the other hand, the proposed EOS, which we expect being implemented in the market and industry during the 2020's (Aurousseau et al., 2013; Grabot, 2012), aims at monitoring and controlling enterprise resources and the operations carried out by those resources. After performing the messages processing by the enterprise resources, the EOS acts as a central orchestrator that intercommunicate and interconnect with the enterprise's IT, Human and Machine dialogues, in order to monitor enterprise components, applications and operations, and allocate resources to required activities based on the interoperability technique (figure 1-C).

3 COMPARISON BETWEEN BOB AND EOS

Best of Breed (BOB) is a leading product/software tool that focuses on providing the best features and functions for one component in the value chain, whether accounting, work orders, sales center, construction management or property management. It is defined as a flexible approach to manage system upgrades and modular application maintenance by utilizing a Service Oriented Architecture which enables easier modular integration. Maintaining multiple systems provides little cross connectivity, which creates maintenance and integration challenges; vendors of best of breed systems are often small organizations that do not understand the requirements of larger organizations, upgrades can get out of sync and cause disruption, total cost of ownership may be high due to multiple vendor support costs, integration with other systems is a highly complex process, and sharing data across different systems may be difficult (Young, 2015).

When the organization expands and requirements multiply, best of breed systems may not be able to handle new requirements, forcing the addition of another system. In this type of scenario, the best course of action is to employ a central orchestrator (EOS) that can ensure the interoperability and integration and can handle most requirements, allowing best of breed systems to handle items requiring focused performance and specialization.

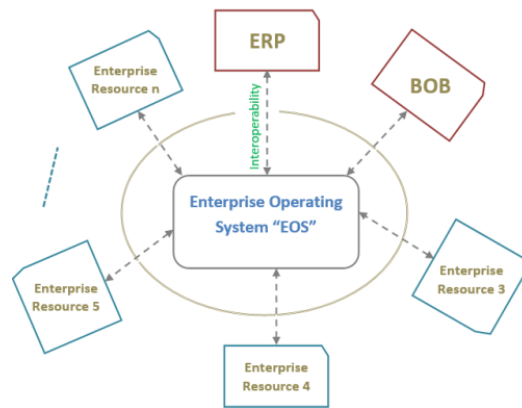


Figure 2: Comparison between BOB and the proposed EOS.

4 POSITION OF EOS BASED ON INTEROPERABILITY

Based on the Enterprise Interoperability Framework, EOS adopts unified and federated approaches by aiming at shifting from the integration to the full federation concept by facilitating 'on-the-fly' interoperations between entities and supporting the communication and transactions between heterogeneous and networked enterprises based on shared business references; it also strives at establishing interoperability through the conceptual and organizational barriers in the four different levels of concerns (data, service, process and business) by adjusting business rules to control operations and tasks, facilitating the alignment and real-time communication with enterprise activities, providing a mechanism to control the

authority of accessing the services, ensuring data bases connection, and providing a data distribution service for implementing on-the-fly information mapping.

The implementation of EOS will use the existing interoperability solutions to support the development of federated and unified approaches of enterprise interoperability. In particular, the HLA architecture will be used to rapidly develop HLA based interface for achieving federated enterprise interoperability.

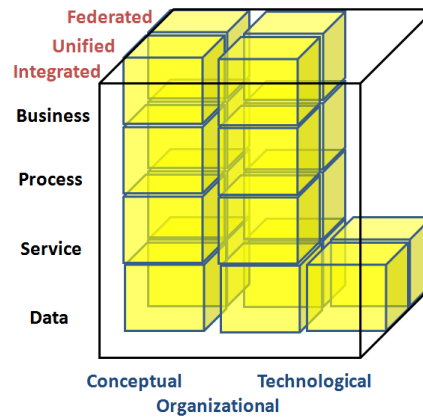


Figure 3: EOS positioned in the Interoperability framework.

5 EOS REQUIREMENTS AND FUNCTIONALITIES

The EOS requirements and functionalities 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” (AMICE, 1993). This Pre-European standard benefits from the result of European Esprit project CIMOSA (Kosanke, 1995). Five categories of requirements representing five basic functions of EOS have been identified as follows (CEN, 1999):

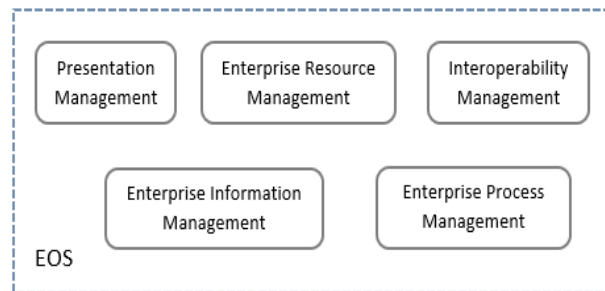


Figure 4: EOS internal components

- Enterprise Resources Management (ERM)
- Enterprise Process Management (EPM)
- Enterprise Information Management (EIM)
- Presentation Management (PM)
- Interoperability Management (IM)

5.1 Enterprise Resource Management (ERM)

Enterprise Resource Management is the efficient and effective deployment and allocation of an organization's resources including:

- Human resources: The sets of individuals who make up the workforce of an organization and business sector
- Machine resources: The types of machines and tools containing one or more parts that use energy to perform an intended action. Machines are usually powered by mechanical, chemical, or electrical means

- IT resources: The collections of physical elements that constitutes computer and communications systems as well as software applications.

ERM provides a real-time and global view of the ‘occupation’ of the resources in a company by monitoring the enterprise resources system-wide (available, occupied, out-of-order...) (CEN, 1999).

Main required functionalities of ERM are:

- Checking the availability of the resources and pre-assigning it when available
- Responding to Event Handling’s request to pre-allocate the critical capabilities required by a process
- Signaling the Process Management that those critical resources are reserved
- Matching the required capabilities to the capabilities of existing available resources
- Selecting the appropriate resource as defined by its capabilities
- Allocating and de-allocating resources
- Releasing the assigned resource inputs
- Signaling the Enterprise Process Scheduling that resources have been assigned or de-allocated
- Managing time and performance management support of enterprise functions and operations
- Ensuring that the right resources are allocated to the right place at the right time.

It is to note that only the resources connected to EOS are to be monitored and managed by EOS.

5.2 Enterprise Process Management (EPM)

One of the important goals of EOS is to support enterprise model execution. An Enterprise Process is a set of sequenced enterprise activities and operations. There are two types of processes to be executed by Enterprise Process Management (EPM) of an EOS (CEN, 1999; Gray & Reuter, 1993):

- Business processes defined by business users will be triggered by events to perform enterprise daily operations. These processes (once created) can be stored in the repository, updated and re-used. Examples of these business processes are: customer order negotiation and confirmation; new product design, production planning, raw material purchasing, finished product delivery, etc.
- Computing processes, the EPM is responsible for interpreting process behavioral, information and resource requirements.

The main required functionalities of EPM include:

- Checking authorization to execute a process
- Sending commands that will trigger the start of processes
- Retrieving process descriptions from the run-time repository
- Recording ending statuses of processes (e.g., done, not done, failed...)
- Notifying the rule sets contained with processes to Event Management Registration
- Activating or terminating the appropriate process
- Responding to Rule Interpretation notifications when a process is notified as terminated
- Providing fault management support for enterprise activities
- Invoking the Condition Monitor service to register conditions that need to be periodically monitored
- Monitoring the progress of process execution and key performance indicators
- Responding to unexpected events and exceptions by requesting Rule Interpretation to invoke the pre-registered exception-handling Enterprise Activity
- Invoking Presentation and Information Management services as necessary to acquire relevant information
- Interacting with the Enterprise Resource Management in order to allocate resources, interpret and receive resource statuses.

5.3 Enterprise Information Management (EIM)

This set of functionalities is fundamental for all enterprise integration infrastructures and platforms. Enterprise Information Management supports information and data exchanges between all internal entities of the EOS as well as external entities connected to the EOS such as enterprise resources and business managers (CEN, 1999; Chen et al., 2015). EIM is required for optimal use of information and its exchange within the organization; for instance, to support decision-making, process execution and day-to-day operations that require the availability of data and knowledge.

The main required functionalities of EIM are:

- Ensuring the centralized management of reference data used
- Automating and arranging master data management
- Excluding data duplicates
- Ensuring and 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 sources and appropriate data storage facility needed for running EOS
- Supporting maintenance of the consistency and integrity of enterprise data
- Supporting data and object queries expressed in a standard format
- Supporting flexible change of the enterprise's information including data definition, configuration control and enforcement of data access rights
- Ensuring information and data confidentiality and security to protect from non-authorized access.

5.4 Presentation Management (PM)

Presentation Management is concerned with the interactive aspects of an EOS that ensure the interactions between internal and external worlds of the EOS. PM produces a set of interfaces which make it easy (self-explanatory) and efficient to organize and coordinate the communication and information flows between Enterprise Resources connected to the EOS and internal entities of the EOS including the Human, Machine and Application Dialogue services (CEN, 1999; Chen et al., 2015).

- Human dialogue service: Refers to the interface between the EOS and human agents (managers and operators) receiving and sending information (command, data, instruction, status...) from and to the EOS. The information can be displayed on a computer screen or a mobile device
- Machine dialogue service: Refers to the interface between the EOS and machine resources (e.g. NC machines, robots, transport systems and any other material transformation devices) receiving and sending information (command, data, status...) from and to the EOS
- Application dialogue service: Refers to the interface between the EOS and computing devices (e.g. computers, applications, databases...) receiving and sending information (command, data, status...) from and to the EOS.

This function 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
- Requesting the involved functional resource to execute the enterprise activity
- Supplying the results of the execution; providing status information on functional entities
- Visualizing an execution trace as a log of completed events; allowing authorized personnel to enter events and activities as defined in the enterprise model
- Modifying the contextual parameters of a simulatable model by the resource at run-time in order to support decision-making, process analysis and exploration of the consequences of alternative scenarios, and interpreting responses from the resources.

5.5 Interoperability Management (IM)

Interoperability Management is a prerequisite feature for an EOS. It is a precondition for a successful implementation of the Enterprise Operating System by allowing different content management systems to inter-operate and ensuring that the EOS can “Talk” to both Business Software Applications and the Embedded Software provided for various machine device controllers and sensors.

The federated interoperability approach (Chen et al., 2015) seems to be the best solution for EOS development. This approach facilitates ‘on-the-fly’ interoperations between entities with a dynamic method through self-adaptation and accommodation. IM must at least provide the following two functionalities:

- On-line interoperability establishment service: It allows a plug-and-play ability to establish the interoperability whenever an external resource is connected to the EOS. IM can automatically detect possible mismatch and perform necessary mappings
- Off-line interoperability engineering service: To establish interoperability between a newly connected resource and the EOS, some human interventions are needed to make necessary changes to solve non-interoperability problems.

The federated approach aims at bridging the gap from interoperability concepts to the implementation of interoperable enterprise systems development. It facilitates and coordinates the communication between heterogeneous distributed information systems of the enterprises by allowing quick interoperability establishment, easy-pass, and dynamic environment update.

It is to note that IM and PM have different roles. PM is concerned with how information is presented (shown) in the display device of a resource (table, graphic, lines...) and how to connect EOS to the resource. IM is to ensure that information exchanged between EOS and its external elements can be interpreted/understood correctly without syntax, semantics and technology incompatibilities.

6 EOS ARCHITECTURES

Three architectures of Enterprise Operating System are designed to meet the related requirements and functionalities. The conceptual architecture defines what core functions are needed in EOS, while the technical architecture specifies how these functions work together to accomplish the EOS mission and the implementation architecture describes what technology and concepts are being used to implement the conceptual and technical architectures.

6.1 Conceptual Architecture

A conceptual architecture is a representation of the main functions of a system from the point of view of its use. It is independent of how it technically works and is implemented using a specific technology. Figure 5 describes the proposed EOS Conceptual Architecture. Unlike ERP, this EOS will be used as a system-wide platform that allows the decisions makers to connect to and communicate through the enterprise systems (hardware, software, network, machines, human operators...) in an effective and efficient way (Chen et al., 2015; Youssef, Zacharewicz, Chen, & Zhiying, 2017).

As shown in figure 5, business users and the three types of resources are outside the EOS. They are connected to the EOS to send and receive commands (or orders) and information (e.g. data, feedbacks, statuses...). To perform enterprise operations (i.e. to design, manufacture and sell required products to customers) in a coordinated way, the resources must be commanded and controlled by business managers via the EOS:

- Human type resources are human actors, operations of whom can be monitored and controlled by the EOS. They are commercial and purchasing agents, product designers, business developers, production managers, shop floor operators, etc.
- IT type resources are computer and communications systems including data processing, storage devices and enterprise applications such as, for examples, MRPII planning software, shop floor

scheduling software, CAD system, sale forecasting software, CRM software, inventory management software, etc.

- Machine type resources are production or transport machines, technical devices, pieces of equipment and tools containing one or more parts that use energy to transform raw material to products such as, for examples, automated/manual transfer lines, conventional and NC machines, robots, etc.

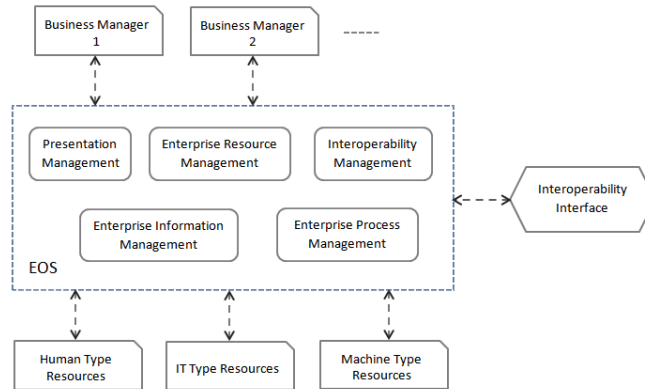


Figure 5: EOS Conceptual Architecture.

Besides the three basic types of resources, two other components need to be connected to an EOS:

- Business managers are not monitored and controlled by the EOS; they are the decision-makers who define what and how enterprise operations will be done, manage the activities and send commands to enterprise resources via the EOS.
- Interoperability Interface allows interoperability engineers to deal with some particular interoperability problems on-line or off-line. It also enables interoperability between EOS internal and external components connected to the EOS.

As shown in figure 5, an EOS is made of five modules providing required services to ensure the functioning of the EOS:

- Enterprise resource management dynamically monitors the connected resources system-wide (available, occupied, out-of-order...), matches the required capabilities to the capabilities of existing available resources; and ensures that the right resources are allocated to the right place at the right time.
- Enterprise process management executes business processes defined by business managers, sends commands triggering the start of processes, records ending statuses of processes and monitors the activity and state of individual enterprise processes as well as some key performance indicators.
- Enterprise information management provides a centralized management of reference data used, automates and arranges master data management, provides transparent access to data sources and an appropriate data storage facility needed for running the EOS and guarantees information and data confidentiality and security to protect from non-authorized access.
- 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.

6.2 Technical Architecture

The technical architecture is still independent of a specific technology for implementation. The figure 6 illustrates and presents the EOS layers' workflow where the enterprise operations are executed and generated through the EOS internal components (Youssef, Chen, Zacharewicz, & Zhiying, 2016; Youssef, Zacharewicz, & Chen, 2016).

At first, each decision maker accesses the General-Purpose and Vertical software interfaces to request a new job from the day-to-day list of activities and operations. The software sends related commands and data to communicate with the EOS front-end interface named Presentation Module in order to execute the requested job.

The Presentation Module then interprets the run-time entities, creates the events which will be registered with their associated information by the Event Registration Component using the Run-Time Repository Service and produces event occurrences for the Event Handling component.

Next, the Event Handling manages the event occurrence priorities, queues and traceability, provides the order identifier and creates the Process Occurrence. The Process Occurrence requests to scheduling from the Process Scheduling component which interprets the process behavior, 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 component.

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: assign resources allocated by the Process Scheduling component, release the involved resources when terminating an activity and notify the termination of the current activity to the Rule Interpretation component.

The Resource Controlling component checks the availability of the resources and pre-assigns them 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 optimization 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 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 appropriate format the current status and the past history of events, allowing authorized persons to intervene manually
- The Machine Dialogue supports the necessary features in order to provide access to the various functional capabilities of 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.

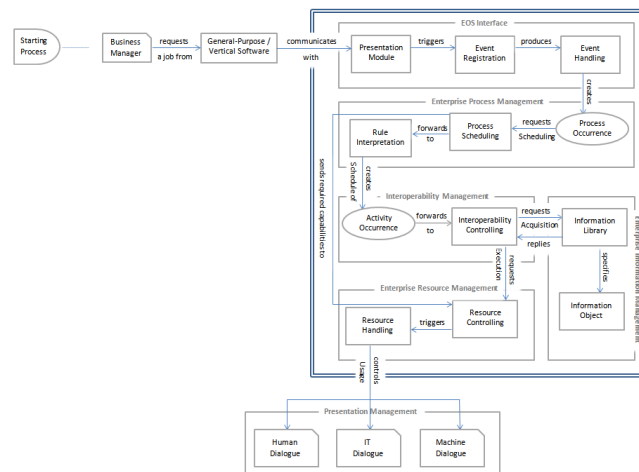


Figure 6: EOS technical architecture.

6.3 Implementation Architecture

The HLA (High Level Architecture) supported by the US Department of Defense is chosen for the Implementation Architecture (IEEE 1516, 2010; Portico, 2013). The main reasons for this choice are 1) its capability to easily enable federated interoperability between heterogeneous applications and 2) its ability to deal with large distributed (simulation) environments. Besides, HLA is also in line with service orientation-based implementation that is one of the main characteristics of EOS principles. Indeed, an EOS is mainly an IT infrastructure made of many services. HLA provides one possible implementation support particularly convenient to EOS. Nevertheless, the implementation of EOS is not restricted to HLA.

As shown in figure 7, all EOS federates are connected together through HLA Federation provided by IEEE in order to publish and subscribe events between federates. This type of structure is used to facilitate inter-federate communications and to support efficient information exchange when participating in a distributed federation execution (Youssef, Zacharewicz, et al., 2016).

The HLA technology allows the EOS federates to communicate data and to synchronize actions among one another based on the federated interoperability features by defining how federates can connect to the RTI, create, join and manage federations, save and restore federation states and define a system to synchronize federates to the same time (Portico, 2013).

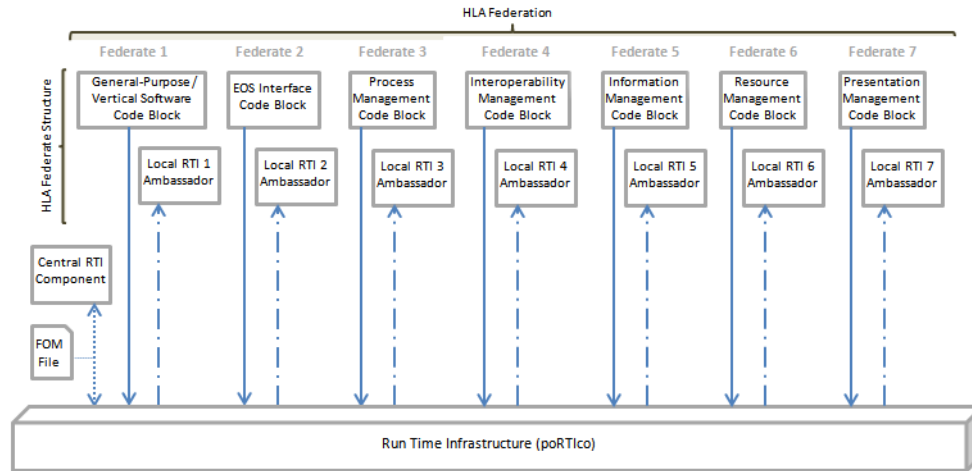


Figure 7: EOS implementation architecture.

poRTico RTI component is the fundamental component presented as an open source, cross-platform used to implement the High-Level Architecture to coordinate and interoperate 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 transmitted between federates. It supports HLA simulation development to greatly expand the use of distributed simulation between all components and federates of the EOS.

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

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" to form a complete federate, and (2) the local RTI component code "LRC" that provides the services for the federate through communication with the RTI executive component and the other federates (poRTico, 2009).

The Federation Object Model "FOM" file describes the set of object classes, attributes and interactions that are shared across the federation. "FOM" contains a copy of the HLA standard Management Object Model, which is a collection of classes and interactions. This file is documented using the HLA OMT format.

Broadly speaking, 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.

7 BANKING SECTOR – CASE EXAMPLE

The illustration case is concerned with an example in banking operations domain. A simulation system in the Banking and Finance environment performing exchange rate definition and update operations is presented, validated 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. Bank exchange rates are constantly changing on 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 Run Time Infrastructure 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.

The EOS internal components are connected together and to the external components through the RTI in order to create a new activity (exchange rate update), analyze this activity, turn it as a process and execute it through special resources and information (Youssef, Zacharewicz, Chen, & Vernadat, 2017).

Each EOS component has a special role during the execution:

- **EOS Interface:** EOS interface takes the charge of transmitting the new requested job (exchange rate update) created by the Bank’s Branch Manager. After that this interface restricts any financial tasks before this modification to prevent any duplicate or errors on the Bank’s daily transactions. Later on, it creates the process of Exchange Rate’s update by defining the related database tables, attributes and specifications.

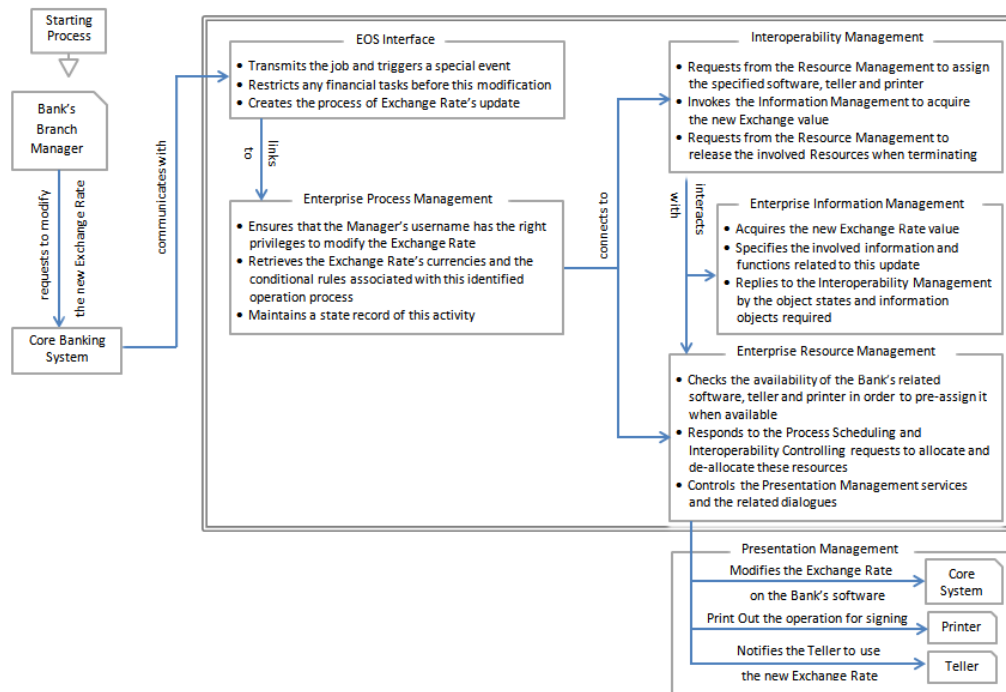


Figure 8: Conceptual and technical architecture of the bank exchange rate update.

- **Enterprise Process Management:** This EOS component is responsible of ensuring that the Manager’s access rights are mentioned as the right privileges to modify the currencies Exchange

Rates. After the confirmation, it retrieves the Exchange Rate's currencies (EURO and USD) and the conditional rules associated with this identified operation process, and after that the "PM" maintains a state record of this activity including the requestor information, attributes modified, date and time.

- Enterprise Interoperability Management: After the creation and generation of the process, the "IM" has two synchronized jobs: 1) requesting from the Resource Management to assign the specified software, teller and printer for exchange rate execution, and 2) invokes the Enterprise Information Management to acquire the new Exchange value. After the process execution, this component requests from the Resource Management to release the involved resources when terminating.
- Enterprise Information Management: When the "IM" notifies the "EIM" about this new task, this EOS component acquires the new Exchange Rate value by specifying the involved information and functions related to this update and then replies to the Interoperability Management with the object states and information objects required.
- Enterprise Resource Management: When the job is ready to be executed, this component checks the availability of the Bank's related software, teller and printer in order to pre-assign it when available. Then, it responds to the Process Scheduling requests to allocate and de-allocate these resources and after that it controls the Presentation Management services and the related dialogues to ensure that the right resources are executing the exchange rate update in the right way and at the right time.
- Enterprise Presentation Management: As the final stage, this component has the role of generating and executing the new job by modifying the exchange rate on the Bank's core banking system. Then, printing the operation approval for being signed by the authorized personnel (Bank's Branch Manager), and finally notifying the teller to use the new exchange rate value.

In the Implementation architecture the seven federates are connected together and they are communicating through poRTico in order to publish and subscribe events between modules. The "Exchange Rate form" of the Core Banking System and "Presentation Management" federates are developed and implemented using the Vb.net programming language on the Visual Studio 2010 platform. The other federates, "EOS Interface", "Enterprise Process Management", "Interoperability Management", "Information Management" and the "Enterprise Resource Management" are developed and implemented using the Java language on Eclipse version 6 platform.

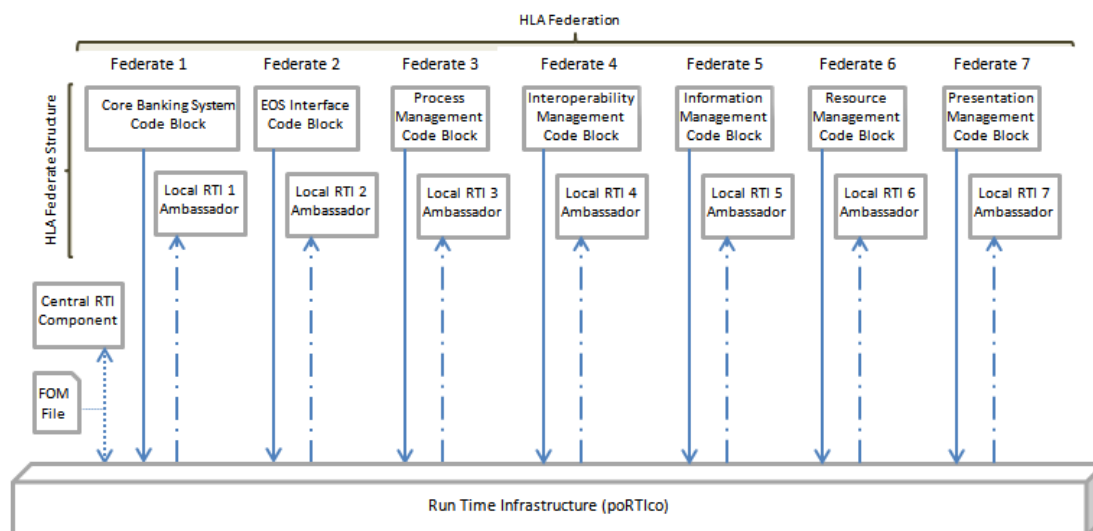


Figure 9: Implementation Architecture of the Bank Exchange Rate update.

As shown in figure 10, the BPMN diagram of this case study generated through Yaoqiang software displays the orchestration and interchange of messages and tasks between EOS components and the external

resources on a clear and organized queue based on the priority of each task (Falcone, Garro, D'Ambrogio, & Giglio, 2017).

As for the technical development and implementation phase, the EOS interface, Enterprise Resource Management, the Enterprise Process Management and the Presentation Management federates 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 (Youssef, Chen, & Zacharewicz, 2017).

- The EOS interface federate contains the “reflectAttributeValues” function and its related parameters used in order to restrict any access for Read/Write on the database and change the status to “ON” as a notification for the Enterprise Operating System that a new job is taking place.
- The Enterprise Process Management federate contains the “getSelectedItem” function that is developed and deployed in order to retrieve the user access rights, compare it and ensure that he has the right privileges for exchange rate update activity. The “actionPerformed” function is mainly responsible of defining and storing the logs and specifications related to the process execution.
- The Enterprise Resource Management federate contains the “receiveInteraction” function that is executed and generated in order to receive all execution’s information including the new value of exchange rate and the assigned resources’ characteristics.
- The Presentation Management federate comprises the “btn_print_Click”, “btn_modify_Click” and “btn_notify_Click” functions that are developed and implemented in order to assign the related resources and execute the job through the related Human, Machine and IT dialogues.

Noting that this poRTico project (RTIBasic) located in the path “C:\workspace\RtiBasic” can execute through the Eclipse platform “Luna\RTIconversion\src\edu\youssefzacharewicz”.

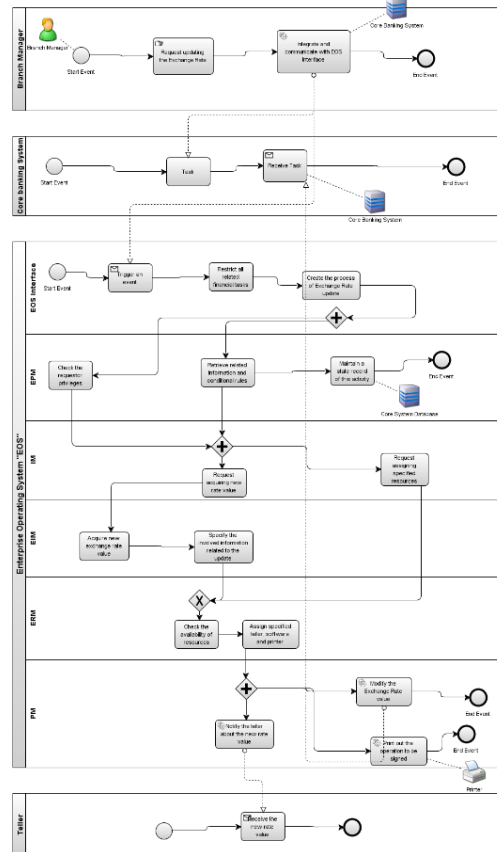


Figure 10: BPMN diagram related to the Banking sector case study.

At first, the “Main” class executes the “run_federate” function in order to start the federation by creating the RTIAmbassador and retrieving the required information from the FOM file. After that the federates join the federation and run the “publish&subscribe” function. Next, the “Conversion” class performs the “Exchange Rate Operation” function. Afterward, the “Conversion_Rate” function draws the interface and query the database in order to retrieve the exchange rate new value. And finally, the “updateattributevalue” function sends the conversion model to the RTIAmbassador in order to notify all assigned resources to perform their tasks and operations. In order to confirm and approve the validity of the Exchange Rate update, this experimental case study has been fully implemented through several heterogeneous environments using the .net and java programming languages based on the Interoperability and Uniformity principles to provide a set of domain-independent APIs used to access capabilities and features and to exchange data between federates using the XML format.

The scenario described above approved the integration of several heterogeneous federates with portTico. Exchange rate has been transmitted, updated and executed through the EOS requirements.

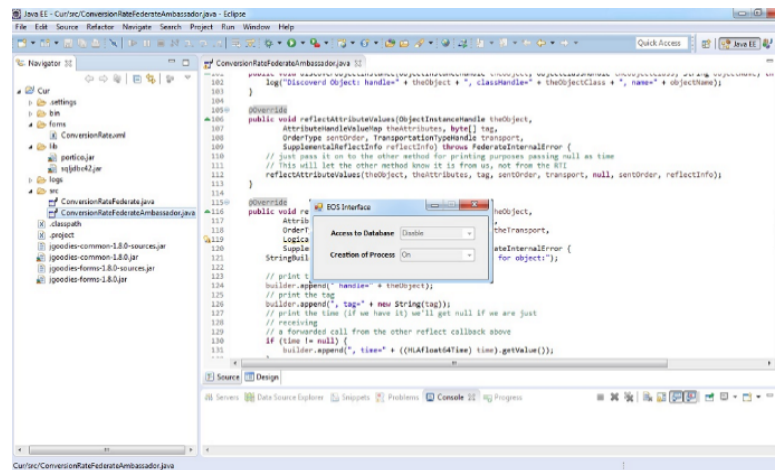


Figure 11: The EOS interface federate integrated with portico.

It must be noted that EOS is also implemented in the manufacturing sector by illustrating how various EOS functions work together to run an enterprise and showing that basic functions of ERP, Workflow and enterprise integration are coordinated by the EOS by commanding the needed operations to produce required products.

8 CONCLUSION

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:

- Provide an alternative for “SMEs”
- Support establishing enterprise interoperability in a heterogeneous environment
- Facilitate re-use of models and re-engineering sub-systems based on models

8.1 Benefits

EOS helps improving the business industry and technology development by:

- Ensuring that all enterprise’s functions and requirements are covered during the EOS execution
- Reducing the enterprise’s systems and deploying only the required heterogeneous applications
- Allowing enterprises (especially the future ones based on IoT and Cyber physical principle) to dynamically monitoring and controlling their operations based on real-time data
- Providing a bottom-up approach with more options for enterprises to implement tailored solution, especially for SMEs

REFERENCES

- AMICE. (1993). *CIMOSA- Open System Architecture for CIM, 2nd and extended revision, Esprit Consortium AMICE*. Springer-Verlag Berlin.
- Aurousseau, M., Ballot, E., Bernard, A., Brissaud, D., Caroly, S., Frein, Y., Grabot, B., & Rocchi, V. (2013, November). *FUTURPROD - Futur Production Systems, Think Tank*.
- Burnson, F. (2015). *Enterprise Resource Planning Software - Buyer Report*. Software Advice™. (Stamford, England).
- CEN. (1999). *prENV 13550 "Advanced Manufacturing Technology - Systems Architecture - Enterprise Model Execution and Integration Services"*. Retrieved from "Association Française de Normalisation." (Paris, France)
- Chen, D., Youssef, J. R., & Zacharewicz, G. (2015, May). *Towards an Enterprise Operating System – Requirements for Standardisation, Proceedings of IWEI 2015*. IWEI Workshops 2015.
- Diercksen, V. (2012, March). *What is ERP?* Retrieved from <http://www.erpssoftwareblog.com/2012/03/do-you-know-the-history-of-erp/>
- Falcone, A., Garro, A., D'Ambrogio, A., & Giglio, A. (2017). *Engineering Systems by combining BPMN and HLA-based distributed simulation*.
- Gable, J. (2002, April). *Enterprise Application Integration*. Retrieved from Questia Trusted Online Research.
- Grabot, B. (2012, November). *GDR MACS, Prospectives STP, Sciences et Techniques de la Production de Biens et de Services*.
- Gray, J., & Reuter, A. (1993). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers (An Imprint Elsevier). (San Francisco, California).
- IEEE 1516. (2010). *"Standard for Modelling and Simulation High Level Architecture". Framework and Rules*. Retrieved from IEEE International Conference on, pp. 1-9, 2011. (New York, USA)
- Kosanke, K. (1995). *CIMOSA - Overview and status*. Computers in Industry 27.
- Linthicum, D. S. (1999, February). *Enterprise Application Integration*. Addison-Wesley Information Technology Series. (Canada).
- Muller, G. (2006). *Distributed Event-Based Systems*. Springer.
- poRTico. (2009). *Developer Documentation*. Retrieved from http://porticoproject.org/index.php?title=Developer_Documentation
- Portico. (2013). *Portico Space: www.porticoproject.org*. Retrieved from <https://portico.space/youre-the-architect>
- Young, J. (2015, July). *AUDIO | Best-of-Breed vs. ERP: What's Best for Higher Ed Today?* The EvoLLLution.
- Youssef, J. R., Chen, D., & Zacharewicz, G. (2017). *Developing and Enterprise Operating System (EOS)- State of the Art*. ICE/IEEE'17 - International Conference on Engineering, Technology and Innovation.
- Youssef, J. R., Chen, D., Zacharewicz, G., & Zhiying, T. (2016). *Developing and Enterprise Operating System (EOS) with the Federated Interoperability Approach*. I3M'16 - International Multidisciplinary Modeling & Simulation Multiconference.
- Youssef, J. R., Zacharewicz, G., & Chen, D. (2016). *Developing and Enterprise Operating System – Requirements and Architectures*. WETICE'16 - 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises.
- Youssef, J. R., Zacharewicz, G., Chen, D., & Vernadat, F. (2017). *EOS: Enterprise Operating Systems*. International Journal of Production Research – 2017.
- Youssef, J. R., Zacharewicz, G., Chen, D., & Zhiying, T. (2017). *Enterprise Operating System (EOS) Framework: Federated Interoperability based on HLA*. International Journal of Simulation and Process Modelling.
- Zacharewicz, G., Chen, D., & Vallespir, B. (2008). *HLA Supported Federation Oriented Enterprise Interoperability, Application to Aerospace Enterprises*. International Simulation Multiconference EuroSISO.
- Zacharewicz, G., Chen, D., & Vallespir, B. (2009). *Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability*.

AUTHOR BIOGRAPHIES

JOSEPH RAHME YOUSSEF is a Senior Lecturer at the American University of Beirut (Lebanon) and ICT Consultant/Auditor at the Lebanese Housing Bank (Lebanon). He obtained his PhD in Computer Engineering from the University of Bordeaux (France) in 2017, specializing in the area of Productive, Signal and Cognitive Engineering. His research focus is on information systems, project planning, distributed simulation, integration and interoperability approaches, model driven approaches. His email address is jy07@aub.edu.lb.

GREGORY ZACHAREWICZ is a Full Professor at the French National Institute for Mines and Telecommunication IMT - Mines Ales (France). His research interests include distributed simulation, semantics aspects, model driven approaches, discrete event M&S, and enterprise modeling. He received his MS (2002) and PhD (2006) in Modelling and Simulation from Aix-Marseille University. His webpage is <https://cv.archives-ouvertes.fr/gregory-zacharewicz> and his email address is gregory.zacharewicz@mines-ales.fr.