



EOS: enterprise operating systems

Joseph Rahme Youssef, Grégory Zacharewicz, David Chen, François Vernadat

► To cite this version:

Joseph Rahme Youssef, Grégory Zacharewicz, David Chen, François Vernadat. EOS: enterprise operating systems. International Journal of Production Research, 2017, 10.1080/00207543.2017.1378957 . hal-01619773

HAL Id: hal-01619773

<https://hal.science/hal-01619773>

Submitted on 2 Nov 2017

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.

EOS: enterprise operating systems

Joseph Rahme Youssef^a, Gregory Zacharewicz^a, David Chen^{a*} and François Vernadat^b

^aIMS, University of Bordeaux, Bordeaux, France; ^bLGIPM, University of Lorraine, Lorraine, France

(Received 24 February 2017; accepted 6 September 2017)

A proposal for developing an enterprise operating system (EOS) for real-time monitoring and control of enterprise operations is presented. The proposed EOS will work on top of enterprise computer operating systems to control and monitor enterprise resources instead of just computer components. A set of requirements and functionalities are first identified. Next, a survey of previous relevant works is presented and results are compared to the requirements. The architectures of the envisioned EOS are then outlined and two examples are presented to illustrate the use of the EOS. The last part draws some conclusions and gives future perspectives.

Keywords: enterprise operating systems (EOS); architecture; infrastructure; interoperability, enterprise operations

1. Introduction

Nowadays, daily enterprise operations are not effectively monitored and controlled. This is due to the lack of a system-wide operating system and insufficient availability of timely real-time data. In the past 30 years, tremendous efforts have been spent to develop enterprise integration, enterprise resource planning and management solutions. These approaches are still not satisfactory because of high cost and long delay to implement them in enterprises. In line with the development of the future generation enterprise manufacturing systems based on internet of things (IoT) and cyber physical system (CPS) principles, this paper presents a proposal to develop a prototype of a so-called enterprise operating system (EOS) that aims at joining various existing heterogeneous approaches together for the sake of complementarity and synergy.

The proposed EOS will behave in the same manner as a computer operating system (OS), but in an enterprise-wide context to monitor and control enterprise operations. The envisioned EOS will execute enterprise models defined by business managers, trigger enterprise operations with dynamically allocated enterprise resources and monitor the status of enterprise resources (Human, Machining, Computing) through various sensing devices and front-ends.

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 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 objective of the paper is to present basic concepts and principles of an EOS, as well as conceptual and technical specifications for the design and implementation of EOS. The conceptual specifications include EOS user requirements and EOS conceptual architecture. From the technical specifications, the EOS technical and implementation architectures will be derived. Both conceptual and technical specifications presented build on previous relevant works presented in the state-of-the-art section.

The paper is structured as follows. After the introduction, Section 2 points out the main deficiencies of existing approaches and states the rationale for developing an EOS. In Section 3, the requirements and functionalities of the EOS are identified. Section 4 is dedicated to evaluate the related works with an extensive survey of state-of-the-art approaches in this area. A specific effort has been made to map existing works to the identified EOS user requirements. Based on the requirements and state-of-the-art results, the conceptual, technical and implementation architectures are outlined in Section 5. Finally, two simplified examples are presented in Section 6 to illustrate the concepts and principles of the EOS in monitoring and controlling daily enterprise operations. Section 7 concludes the paper and points out some future perspectives.

*Corresponding author. Email: david.chen@ims-bordeaux.fr

2. Problem statement and rationale

Successful business organisations are becoming increasingly complex and dynamic due to the arrangement, coordination and management of complex computer systems, middleware and services. These enterprises are redesigning their structures and business processes to perform enterprise application integration through innovative approaches and techniques in order to accommodate and implement the increasing complexity of services, data orchestration and distribution methods. Since 30 years, there have been many initiatives to develop enterprise integration platforms and infrastructures such as CORBA, EAI, etc. (Linthicum 1999). However, although important achievements have been made, these initiatives failed to provide satisfactory solutions to efficiently integrate heterogeneous enterprise software applications, especially for SMEs. Enterprise integration is still seen as a complex, costly and time-consuming task (Chen 2005; Chen, Doumeingts, and Vernadat 2008; Vallejo, Romero, and Molina 2012; Weichhart et al. 2016).

On the other hand, several software packages provided by ERP vendors (e.g. PeopleSoft, SAP or JD Edwards) and other IT products deliver improved enterprise application integration by offering an integrated suite of applications to perform standard business functions. Until recent years, packaged-software vendors concentrated on one application or a suite of applications that they automated without regard to other applications that a company might have in its systems portfolio. This solution may constrain the business due to the top down ‘enclosing’ methodology. Typically, existing ERP packages are becoming enormous in size and fairly complex, which makes the implementation and the updates very time consuming, involving high costs and disruption of on-going everyday enterprise processes and operations (Gray and Reuter 1993; Burnson 2015).

A different approach may take place by setting loosely coupled connections between enterprise software systems following the idea of federated interoperability with only one unique system-wide platform, called ‘enterprise operating system’ (EOS) as proposed in (Chen, Youssef, and Zacharewicz 2015; Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016). Moreover, developing an EOS can also be seen as one of the key steps towards the future generation of manufacturing systems to make real-time enterprise operations monitoring and control a reality. From this point of view, the EOS approach presents the better perspective for the future in line with the IoT and CPS principles (Weichhart et al. 2016). Furthermore, developing an EOS would allow the reconciliation of two different but complementary initiatives for enterprise integration and operation management: IT platforms/infrastructures and ERP-based application packages (Zarató 2013).

Compared to ERP, EOSs have similar objectives but adopt a different approach. ERP is a top-down integrated approach that includes in its framework the main enterprise applications and operating functions. EOS is a bottom-up federated approach only providing enterprise operating system functions that allow diverse heterogeneous and multi-vendor applications to be connected and to operate together. From this point of view, EOSs provide more options for enterprises with tailored solution, especially for SMEs. At a lower cost compared to a full ERP solution, a SME equipped with an EOS will deploy and install only required (potentially heterogeneous) enterprise applications.

It is important to note that once an EOS is implemented in a company, an ecosystem could then be developed to work with a variety of enterprise applications compatible with this EOS, just like what happened with Apple’s ecosystem to iOS and Google’s one to Android. To make this happen, the envisioned EOS approach needs to be recognised as a standard so that enterprise solution vendors develop their applications according to the specifications of the EOS.

More precisely, an EOS can be defined as ‘a set of IT services supporting execution, control and monitoring of enterprise operations by means of message exchange among enterprise resources, model enactment and interoperability capabilities’. Table 1 shows the analogy between a computer OS and an EOS to highlight the similarities and differences of these two concepts.

In the context of industry 4.0, the machines that will work as autonomous production systems, and more generally artificial agents will also need to access collective information (and commands) to efficiently operate.

Noting that implementing an EOS in a company does not mean that the EOS will monitor and control all the resources and all operations carried out in the company. This depends on the business of each specific company and the degree of automation of that company. Broadly speaking, only operations defined in a business process model and the resources required in the model can be controlled by the EOS.

3. Requirements and functionalities

What are the main functions/capabilities that an EOS must provide? This question is closely related to another one: what are the requirements of an EOS? Therefore, this section identifies a set of the requirements and functionalities for the design of an Enterprise Operating System. They are inspired from and based on early works on the integrating infrastructure (IIS) defined in the CIMOSA architecture (Kosanke 1995) and the pre-European standard known as prENV

Table 1. Analogy between computer OS and EOS.

	Computer OS	EOS
Purpose	To control and monitor IT components/resources connected to OS	To control and monitor enterprise resources connected to EOS
Where implemented	• Local • Inside a computer	• System-wide • On top of computers
Who use	Users of a computer	Business managers and enterprise system operators
Resource monitored /controlled	• Hard disc reader/writer • Mouse/keyboard • Printer/scanner • Monitors • Data storage devices • etc	• Computing devices • Machining devices (NC machines, robots, transport systems, autonomous production systems, artificial agents etc.) • Humans (managers, designers, shopfloor operators)
Operations monitored/controlled	• Input commands • CPU computing operation • Print/scan • Read/write data • etc	• Input commands • Engineering operations • Manufacturing operations • Management operations • etc
Processes	• Internal processes to coordinate and synchronise operations of internal and external components	• Business processes • EOS internal processes to coordinate and synchronise operations of EOS services

13550 'Advanced Manufacturing Technology – Systems Architecture – Enterprise Model Execution and Integration Services' (CEN 1999). This Pre-European standard itself benefits from the result of the European project CIMOSA (AMICE 1993; Kosanke 1995).

The mission of an EOS is to support at least three basic operational aspects of an enterprise: interoperation, monitoring and control. To do so, five categories of preliminary requirements representing five basic sets of functions of an EOS have been identified as follows (CEN 1999):

- Enterprise resource management (ERM)
- Enterprise process management (EPM)
- Enterprise information management (EIM)
- Presentation management (PM)
- Interoperability management (IM)

3.1 Enterprise resource management

The ability to send commands to appropriate available resources to execute operations and possibly to receive feedback is essential for an EOS. ERM aims at efficient and effective deployment and allocation of organisation resources including:

- Human resources: i.e. the sets of individuals (or human agents) who make up the workforce of an organisation and business sector.
- Machine resources: i.e. the set 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: i.e. the collections of physical elements made of 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 (e.g. available, occupied, out-of-order ...) (CEN 1999). The main required functionalities of ERM are: checking, searching, reporting and selecting the available 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.

It must be noted that only the resources connected to the EOS are to be monitored and managed by the EOS; for example, those involved in supporting defined business processes. IT and automated machine resources must be connected directly to the EOS. Human resources are connected to the EOS via appropriate mobile devices and interfaces.

3.2 Enterprise process management

One of the important goals of EOS is to support enterprise model execution (Weichhart et al. 2016). An enterprise process is a set of sequenced enterprise activities and operations. There are two types of processes to be executed by EPM of an EOS:

- Business processes: these are processes defined by enterprise managers to fulfil engineering or manufacturing missions of the enterprise in order to meet market demands. These processes are external to the EOS.
- Computing processes: these are processes defined by EOS developers to fulfil coordination and synchronisation activities of EOS. These processes are internal to the EOS to ensure EOS functioning.

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.

As for computing processes, the main required functionalities of EPM include: sending commands that will trigger the start of processes, recording ending statuses of processes (e.g. done, not done, failed ...); monitoring the progress of process execution and key performance indicators and interacting with the ERM in order to allocate resources, interpret and receive resource statuses (Gray and Reuter 1993; CEN 1999).

3.3 Enterprise information management

This set of functionality is fundamental for all enterprise integration infrastructures and platforms. EIM 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 (Chen, Youssef, and Zacharewicz 2015; Youssef et al. 2016). EIM is required for optimal use of information and its exchange within the organisation; 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 centralised management of reference data used; automating and arranging master data management; excluding data duplicates; ensuring and maintaining system-wide consistency as well as integrity of enterprise data exchanged; providing transparent access to data sources and appropriate data storage facility needed for running EOS and ensuring information and data confidentiality and security to protect from non-authorised access.

3.4 Presentation management

PM 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 organise 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, Youssef, and Zacharewicz 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.

3.5. Interoperability management

IM 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, Youssef, and Zacharewicz 2015; Youssef et al. 2016) 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.

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.

4. State-of-the-art

This section provides a state-of-the-art review of the existing works that have been carried out since the 1980s to support enterprise integration and interoperation. These approaches are analysed in relation to the set of EOS requirements listed in the previous section to assess their effectiveness with respect to EOS. A comparison study is also provided so that their relevance to EOS can be appropriately determined.

4.1 Overview of main works

In the mid 1980s, under the framework of the European ESPRIT Programme, the AMICE consortium launched the CIMOSA project (CIM Open System Architecture) to develop Computer Integrated Manufacturing (CIM) solutions (AMICE 1993). One of the results of the project is a proposal of an 'Integrating InfraStructure' (IIS) for CIM. This is the first ambitious initiative to combine enterprise model execution and enterprise information integration in one platform to develop large integrated manufacturing systems. Although the proposal did not lead to any commercial product, its concepts and principles have been reused in several initiatives after the project.

At the beginning of the 1990's, inspired by the IIS of CIMOSA, the first version of the Common Object Request Broker Architecture, known as CORBA, was proposed as a 'standard' in 1991 by the object management group (OMG). The final version is CORBA 3.3 published in November 2012 (OMG 2012). CORBA was designed for facilitating and enabling the communication of systems deployed on diverse platforms, written in different languages and implemented on different operating systems and models. It offered benefits such as inheritance, information hiding, reusability and polymorphism. However, CORBA can be better considered as a simple enterprise application integration platform rather than a true enterprise operating system (Bryan, Dipippo, and Fay-Wolfe 2005). For instance, it does not provide functionality for enterprise model execution.

In the late 90's, CEN TC310/WG1 (a working group of the European standardisation committee CEN) developed the European pre-standard ENV 13550 on the basis of the CIMOSA IIS principles. The main objective of ENV 13550 is to support system-wide business process monitoring and control (Gray and Reuter 1993; CEN 1999). It 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 while integration of diverse heterogeneous applications is not as well supported as in CORBA (Shorter 1997). Interoperability issues were not addressed in ENV 13550.

Going one step further than CORBA, Enterprise Application Integration (EAI) platforms are integration frameworks made 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 (e.g. SCM applications, ERP systems, CRM applications, payroll ...) (Linthicum 1999). An EAI architecture is typically based on four components including a centralised broker (or hub) that handles message routing, security services, communication services and connectors.

The latter are used to interface each application to the EAI hub so that it can communicate with the other applications. The EAI approach focuses on enterprise application integration by means of normalised message exchange (preferably XML). It was not dedicated to enterprise operations monitoring and control as foreseen for EOS. In addition, EAI products are based on tight integration approaches, leading to so-called monolithic architectures, rather than on loosely coupled interoperability, which is more suited for EOS because of its agile nature (i.e. components can be easily added, removed or modified in the architecture).

In parallel to various initiatives for developing integration architectures and platforms, enterprise model execution also gained wide attention in the mid 1990's with workflow management (WfM). One significant promoter of the workflow management technology was the Workflow Management Coalition (WfMC), which is a non-profit organisation formed to define standards for the interoperability of Workflow Management Systems. The concept of workflow is inspired from Business Process Model execution as defined in CIMOSA (AMICE 1993). It focuses mainly on process definition file interchange using the 'XPDL' standard, which presents a process design format for storing the visual diagram and process syntax of business process models (Hollinsworth 1994). Like ENV 13550, the aim of WfM is process model execution, which is one of the main functions of an EOS.

It is worth mentioning that during the 1990's, the Open Software Foundation (OSF), also a non-profit organisation, has also developed a widely implemented platform for distributed computing called Distributed Computing Environment (DCE). DCE is a middleware platform for heterogeneous systems, used to allow different operating systems to distribute processing and data across the enterprise by providing a coherent environment for developing and running applications. It supports the sharing of information and provides services to mask the complexity of multivendor network environments (Turner 2012).

Also in the same period, another important initiative under the frame of ISO/IEC was the development of a Reference Model for Open Distributed Processing (RM-ODP). This international standard was published in 1990 to describe and build widely distributed systems and applications in a multi-vendor environment (ISO/IEC 2009). The ODP framework integrates aspects related to the distribution, interoperation and portability of software systems, so the operating systems, programming languages, databases and management systems are transparent to the user, and it manages also complexity and maintains consistency of information (Zurawski 2004). ODP and DCE are similar approaches aiming at supporting stem-wide distributed communication and sharing of information. This is a core functionality to be provided by EOS. However, both RM-ODP and DCE do not support enterprise model execution and they are dedicated to IT systems and not aimed at enterprise operations monitoring and control.

Around the years 2000–2002 with the development of the service-oriented approach, the concepts of enterprise service bus (ESB) was proposed (Vollmer 2011). An ESB is a software platform supporting interoperability between heterogeneous environments by routing XML messages using a message bus or message-oriented middleware (Chappell 2004). The concept has been developed in analogy to the bus concept found in computer hardware architecture combined with the high-performance computer operating systems and used for designing and implementing communication between mutually interacting computing hardware components (Flurry and Clark 2011).

4.2 Comparison

The comparison is carried out on the approaches just presented. The criteria used for comparison are based on the five categories of requirements and desired sets of EOS functionality as specified in Section 3.

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 2 shows the comparison.

Table 2. Comparison of existing approaches against EOS requirements.

	ERM	EPM	EIM	PM	IM
CORBA	+	+	+++	+++	+
ENV 13550	+++	+++	+++	+++	-
ESB	++	+	++	-	-
WfM	++	++	+++	+++	++
EAI	++	++	+++	+	+
OSF/DCE	-	+	+++	+++	++
ODP	+	-	+++	+++	++

Concerning 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 WfM deal with other non IT enterprise resources such as 'human' and 'machine'. Table 2 gives the evaluation results for the various approaches regarding relevance to enterprise resource management. CORBA provides a distributed resource management service that monitors slightly the information about resource usage and availability and ensures that adequate resources are available for use when and where they are required (Zarató 2013). ENV 13550 also allows monitoring status of resources (e.g. available, occupied ...). It commands connected resources by sending execution orders through process management service. An ESB contains a module called 'Request Handler' to connect users to the ESB and to different parts of the framework. It forwards the data source requests from the user to the service manager and uses the selection and restriction processes to find the best data source for each resource (Rademakers and Dirksen 2008). WfM includes a 'role coordinator' component which is responsible for managing the resources that can perform a particular role and their related state information (WfMC 1999). Concerning EAI, it allows resource access to products and information and enables them totally to control business processes involved in interactions between different business applications by matching the needs of each resource to the process available and compatible (AIIM 2000). While in DCE the resource management is very limited, there is no single entity that has full control of the resources including the availability, the status, the location, the information and the properties of each resource (Moon and Lee 2006).

Regarding Enterprise Process Management, most of the existing approaches include Process Management. However, ENV 13550 and WfM provide the most interesting concepts and principles to define EOS, especially in the sense that EOS is focusing on the business process execution defined by the decisions-makers. CORBA includes a central building block called Object Request Broker (ORB), which delivers the requested objects and returns the results to the resources even if the process is executed on a different platform. It can determine if a process failed for reasons in case of not receiving an exception (Henning 2006). ENV 13550 aims at dynamically controlling the process model run-time execution by sending commands triggering the start of processes, recording the ending statuses, sending required capabilities to Resource Management to find/allocate the available qualified resources. An ESB includes an engine controlled by the process description that coordinates the collaboration of the services connected to the bus in order to execute the related processes (Sethi, Raynaud, and Faure-Vincent 1995); it contains process configuration elements such as endpoints, fault handlers, process mapping and tracking in order to define how a process flows between services. WfM engines have a 'process coordinator' component that manages the description of a particular process and creates a new case coordinator for new orders instantiating and transferring the plan to that object (Sethi, Raynaud, and Faure-Vincent 1995). In EAI products, a 'Process Layer' is used for automated processes and to ensure the proactive control of the entire process from instantiating a predefined workflow type all the way to its completion (Gable 2002). Moreover, it provides a business process-oriented solution that is configurable across applications and that supports automated deployment by integrating and distributing business processes. Concerning DCE, it has the 'Access Control' model, which provides an object level protection and auditing of each process. The user library allows processes to be added or deleted by sending messages to the related resource and permits to a process to start or terminate using the operating system services (Fagg, London, and Dongarra 1996). ODP also includes a basic concept defining the rules that govern the collective behaviour of a set of processes, the period in which a process exists and its terminating behaviour (Chang 2005).

Concerning EIM, managing system-wide information is essential in enterprise integration and all existing approaches can support the EIM with different levels. Nevertheless, due to the heterogeneous data sources in an enterprise, interoperable solutions are more suitable and fundamental to support this function. CORBA includes a local information collection component, which records all information including the progress of activities, status of processes and data exchanges. It manages the information in the system repository, with all information stored about end-to-end processes and resources (Wainer, Filho, and Madeira 2000). ENV 13550 provides access to system-wide information and data interchange facilities from all documents between enterprise resources connected to the Enterprise Operating System. It aims at ensuring data confidentiality and security to prevent any unauthorised access, requesting to provide required services for establishing interoperability between non-interoperable data sources and supporting flexible change of information. ESB solutions encapsulate the information offered by their component applications in a meaningful way using an 'Enterprise Message Model' which routes the information transmitted and received by the ESB to the appropriate application after transforming the message into a format that the application can interpret. They also monitor and control routing of information exchange between services (Chappell 2004; Rasta 2013). Concerning WfM, workflow engines include the 'case coordinator' component that centralises all information concerned with each case by managing the activities of that case, detecting failures, coordinating its recovery procedures, answering to queries about the case and notifying the process coordinator when a case is closed. EAI-based commercial tools (such as Tibco for example) can also ensure the proper routing and transformation of messages between applications in multiple systems (Linthicum 1999). DCE provides an end-to-end view of business information by providing common definitions of enterprise data

to be integrated in the internal system architecture. In the case of ODP systems, they interconnect the distribution and processing of information and support the exchange of data across organisations in the enterprise. They guarantee a backup of information to recover in case of failure and allow resources to perceive immediate access to data.

The PM is a necessary function of EOS to interact with all resources and business users connected to EOS. ENV 13550 presents the most interesting concepts because it allows dialogue with all types of enterprise resources (human, machine and IT). CORBA uses the middleware services siting in the middle of the platform and application layers which are responsible for PM by providing a set of application programming interfaces in order to allow resources to operate using any type of platforms. ENV 13550 also has PM providing 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 Application dialogue services. ESB has no PM but an Eclipse-based graphical development environment for designing, testing and running ESB flows. It consists of editors for development and a management console (Wyszkowski 2011). As for WfM, it contains a 'worklist handler' that manages the interaction between the user and the work list maintained by a workflow engine. EAI platforms only have a front-end to cluster applications in order to provide a single consistent access interface, to ensure the consistency of information and to simplify and automate business processes (CEITON 2014). However, the EAI user interface is unable to handle more than few screen interfaces at a given time, which may prove unstable if the mechanisms are not set up and developed correctly (Jin 2009). DCE contains a graphical user interface-based management environment that allows a secure remote management of any resource and process in an enterprise network and helps resources to monitor all aspects of the system and its applications. ODP possesses an interface that interacts as an abstraction of the behaviour of the workflow that consists of a subset of the resources with a set of processes to define when the identified interactions can occur.

Last but not least, interoperability is a key function to allow diverse heterogeneous enterprise resources connected to the EOS to perform enterprise operations. Interoperability in existing approaches is not developed to a mature and satisfactory level. A set of interoperability utility services needs to be implemented to support EOS. CORBA includes the ORB Interoperability Architecture, which is defined as a tool for the enterprise system environment in which a single interface can be developed for each resource having multiple processes that can be used (Omicini, Petta, and Pitt 2003). ENV 13550 is not designed for supporting interoperability between enterprise resources. ESB establishes an IM System to connect platforms to each other and organise the share of resources (Konstantas et al. 2006). WfM possesses the 'synchronisation activity' feature that organises the workflow between resources by managing the 'and-joins' and 'or-joins' notifications. It also defines the 'Gateway Activity' feature, which is responsible of the bidirectional workflow data, control and process definition conversions between different workflow management systems (Saha, Mukherjee, and Bandyopadhyay 2011). EAI tools provide utilities and infrastructure services for integration of enterprise applications that enables information sharing and business processes. However, data sources cannot normally operate in heterogeneous environments and they are facing problems of communication and execution related to their respective frameworks (May 2001). DCE facilitates the development and deployment of interoperable applications in the distributed heterogeneous networked environment and it also helps for the integration of legacy applications with new distributed platforms (Xu 2014). ODP systems are based on open solutions to provide general connections among software entities and ensure the cooperation and sharing of information (Zurawski 2004).

4.3 State-of-the-art summary

This section presents a summary of the relevance and insufficiencies of the existing approaches based on the EOS requirements and functionalities.

Considering enterprise resource management, most of the existing approaches only focus on IT resource management (monitoring and control). However, ENV 13550 and WfM also deal with other non IT enterprise resources such as 'Human' and 'Machine' and allow discovering and matching required capability to existing available resources. Referring to Enterprise Process Management, most of the existing approaches deal with Process Management. However, ENV provides the most interesting concepts and principles to manage EOS processes defined by the business users and orchestrated to ensure internal working of platforms. Regarding EIM, existing approaches all support this function to some extents but due to the heterogeneous data sources, data interoperability is not yet well implemented. Concerning the PM, ENV 13550 presents the most interesting concepts, as it allows to dialogue with all types of enterprise resources (human, machine and IT). For Interoperability Management, existing approaches focus on Enterprise Integration rather than loosely coupled interoperability.

In conclusion, the existing works and models are all relevant for the design and development of an EOS but they do not support and cover all the requirements and functionalities needed by EOS. Each of them has its particular objectives with specific focuses. CORBA and EAI are seen more as enterprise application integration platforms rather than

Enterprise Operating Systems. Compared to CORBA, ENV 13550 moved a step towards an EOS. Unfortunately, the work of ENV 13550 is discontinued. It must be noted that service orientation has gained interest in the middle of the 1990's and has been used as a basic principle to develop enterprise integration and interoperability platforms. The ESB developed for this purpose cannot be directly used as an EOS but they provide interesting concepts and principle to develop and build the EOS. Last but not least, and at a technical level, some other approaches are also interesting to take into account to develop EOS technical architectures. We can mention as an example HLA (High Level Architecture) that allows building federated interoperability capability in the EOS (Knight et al. 2002).

5. EOS architectures

This section presents the conceptual, technical and implementation architectures of an enterprise operating system that are designed to meet the requirements and functionalities identified in Section 3. 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.

5.1 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 1 describes the EOS Conceptual Architecture proposed. Unlike ERP, this EOS will be used as a system-wide platform that allows the decision-makers to connect to and communicate through the enterprise systems (hardware, software, network, machines, human operators ...) in an effective and efficient way (Leadbeater 1999).

As shown in Figure 1, 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 and manufacture required products from 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, MRP II 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.

Besides the three basic types of resources, two other components need to be connected to an EOS. They are named Business Manager and Interoperability interface.

- 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 is a universal repository for integrated applications allowing 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 1, an EOS is made of five modules providing required services to ensure the functioning of the EOS:

- PM is a set of services with appropriate interfaces that allow business users and other enterprise resources to be connected to the EOS and to receive/send commands and information (generically called messages). It interprets needed communication protocols, organises and displays the information (data, commands ...) in an appropriate form (text, table, graphics ...) to the elements connected to EOS according to the usage of the information.
- IM is a set of services that provide necessary mapping between heterogeneous resources to make them interoperable through the EOS. It is responsible of removing barriers of interoperability between the elements, such as for

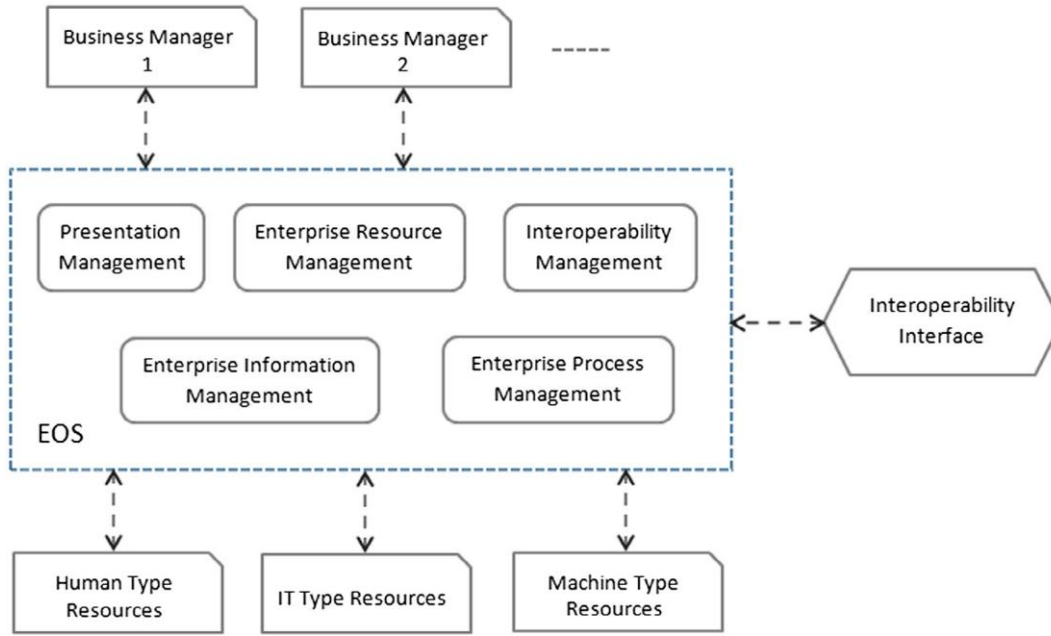


Figure 1. EOS conceptual architecture.

example conceptual barriers (syntax and semantics) and technological barriers related to the use of heterogeneous computing devices.

- ERM 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.
- EPM 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.
- EIM provides a centralised management of reference data used, automates and arranges master data management, ensures and maintains consistency of data with the system-wide run-time name space, manages the message flow and information process, 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- authorised access.

5.2 Technical architecture

A technical architecture describes how components or services of a system interact to fulfil the functions defined in a conceptual architecture. The technical architecture adds details to conceptual architecture but it is still independent of a particular technology for its implementation. Figure 2 illustrates and presents the EOS internal workflow where the enterprise operations are executed and generated through the EOS internal components (Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016).

To start, a business manager needs to connect to the General-Purpose and Vertical software interfaces to ask for a new task from a list of daily activities and operations. The software sends related commands and information to create a connection with the Presentation Module acting as an EOS's front-end interface to execute the requested activity.

The Presentation Module at that point interprets the run-time entities, generates the events to be saved with their associated information by the Event Registration Component using the Run-Time Repository Service and creates occasion events for the Event Handling component. Noting that the Run-Time Repository Service is integrated and implemented with EOS components in order to describe the sequence of steps required to execute the registered events and to maintain message history information of each new event. Once the integration is deployed, the run-time repository contains the metadata required to generate the event instance as well as the event occurrences and related specifications (Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016).

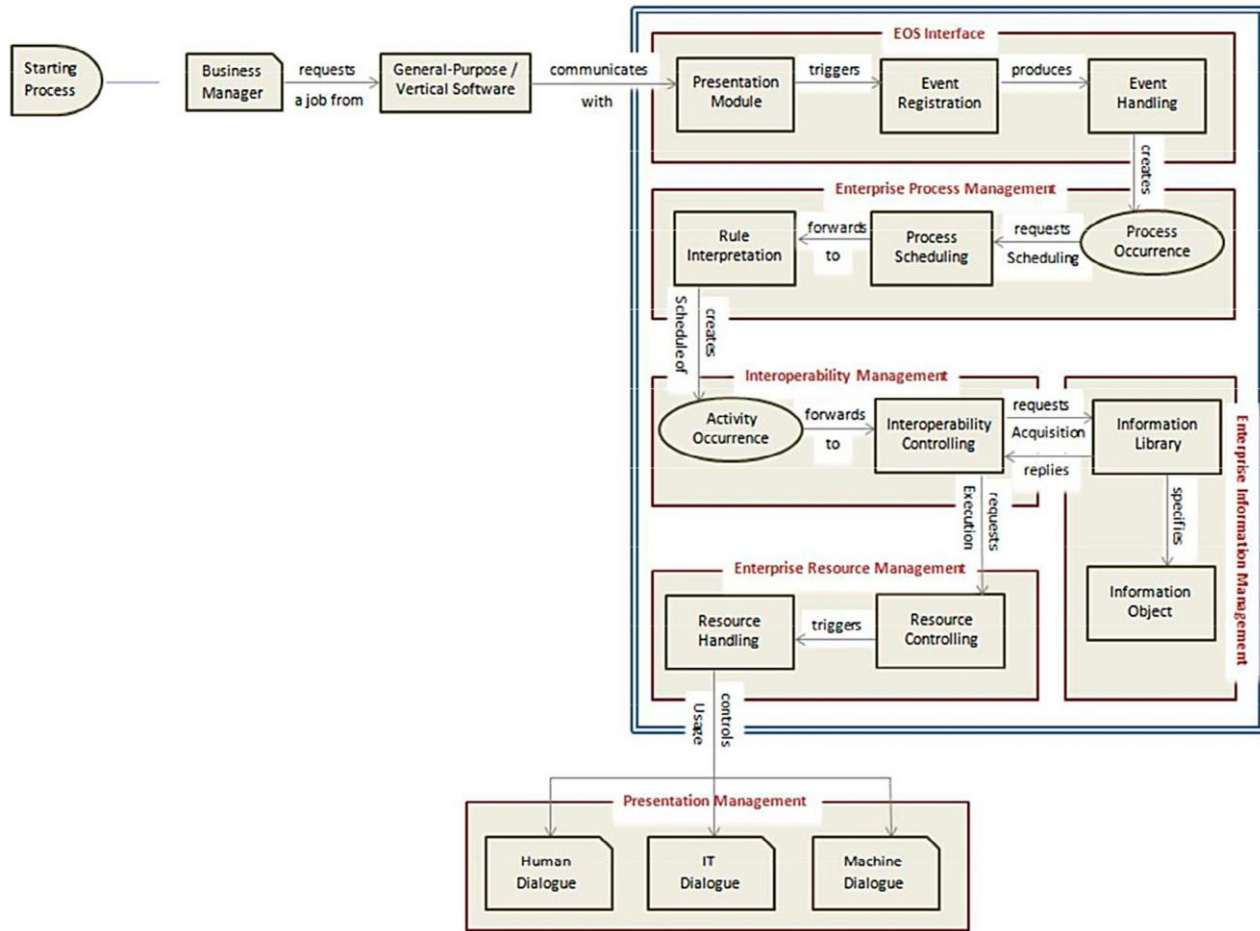


Figure 2. EOS technical architecture.

Next, the Event Handling controls the event occurrence priorities, queues and traceability, sets the order identifier and produces the Process Occurrence. The Process Occurrence asks for scheduling from the Process Scheduling entity that deciphers the procedure conduct, data and resource requirements. This sub-service examines the authorisation to generate the process, extracts process details from the Run-Time Repository, requests the ERM to allocate an available resource according to required capabilities and forwards the details to the Rule Interpretation component (Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016).

Afterward, the rule interpretation entity provides the required functionality in order to retrieve the conditional rules related to the recognised enterprise process, safeguards a state log of all enterprise operations, and reacts to identified events in order to initialise and/or terminate the activity (Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016).

Therefore, and after being generated by the Rule Interpretation, the Activity Occurrence schedule is transmitted to the Interoperability component. This component has both roles:

- (1) Invoking the EIM in order to obtain the object states and designate the information objects required.
- (2) Requesting from the ERM in order to assign resources allocated by the process scheduling component, release the involved resources when ending an activity, and inform the Rule Interpretation component the end of the current task.

The Resource Controlling component is responsible of, verifying the availability of the resources in order to pre-assign them when accessible, replying to the Interoperability Controlling demands in order to assign agents, and responding to the Process Scheduling requests for allocating and de-allocating resources. The Resource Handling component selects the appropriate resource after matching the abilities required and by taking into consideration the time, performance and priority.

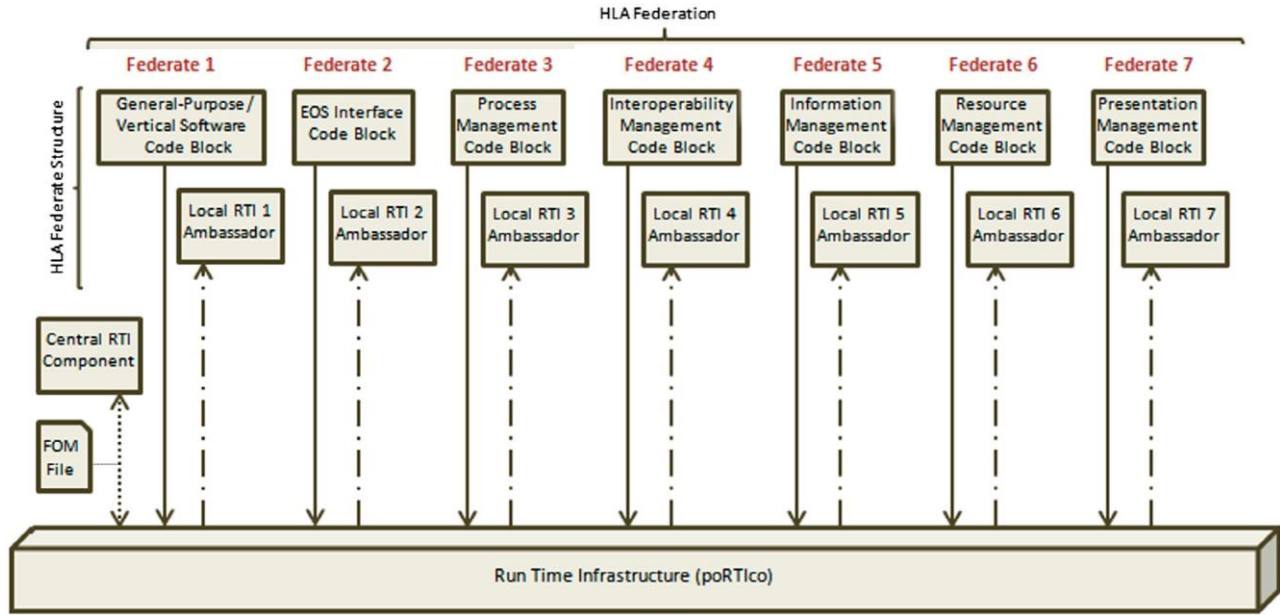


Figure 3. EOS implementation architecture.

The PM services are controlled by the Resource Handling Component for handling Human, Machine and IT Dialogues (Youssef, Zacharewicz, and Chen 2016; Youssef et al. 2016).

- The Human Dialogue provides functionality for displaying in appropriate format the present status and the previous history of events, enabling authorised persons to intervene manually in order to adjust the contextual parameters at run-time.
- The Machine Dialogue supports the necessary options and features in order to allow accessing to the various functional capabilities of the machine. It provides the required functionality for receiving and interpreting responses from the device.
- The IT Dialogue provides functionality for examining application programme interfaces to determine its capabilities, and offering support for the integration of the functional entities implemented by existing IT application programmes.

5.3 Implementation architecture

An implementation architecture describes how to operationalise a system. Figure 3 shows the EOS Implementation Architecture by defining the technology and concepts used to implement the technical architecture. For a given technical architecture, various different technologies can be used and combined for implementation. In this proposal, the HLA (High Level Architecture; IEEE 1516 2010) supported by the US Department of Defence is chosen (Knight et al. 2002). The main reasons for this choice are (1) its capability to facilitate federated interoperability between heterogeneous applications and (2) its ability to synchronise 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 authors do not intend to restrict EOS implementation to HLA.

As shown in Figure 3, all EOS federates are connected together through a 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.

The HLA technology allows the EOS federates to communicate data and to synchronise 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 synchronise federates to the same time (poRTico RTI 2009).

Each federate is made of two blocks: (1) the Code Block, which contains the Federate Ambassador which execute using pure virtual functions and abstract base classes to send messages and requests to the RTI (Run Time Infrastructure)

and (2) the Local RTI Ambassador, which provides the services and protocols for the related federate by connecting to poRTIco RTI application.

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 facilitate the use of distributed simulation between all components and federates of the EOS (poRTIco RTI).

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

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.

It must be noted that this Implementation Architecture is fully implemented using heterogeneous development languages (Java, Vb.net, SQL ...) 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. Additional detail can be found in one of the case studies presented in Section 6.

6. Case studies

This section provides two simplified examples to illustrate concepts and principles of the proposed EOS. The two examples deal with two different application domains: manufacturing and service. The service case deals with the banking sector. It is implemented using the HLA technology for the purpose of proof of concept from the implementation point of view. The manufacturing case focuses on conceptually illustrating how enterprise operations are monitored and controlled through the use of the EOS.

6.1 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 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. Bank exchange rates are constantly changing on every business day based on current market conditions. Figures 4 and 5 show the EOS Conceptual, Technical and Implementation Architectures of the Bank daily Exchange Rate update operations. The implementation architecture is designed using HLA technology.

As shown in Figure 5, 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 'PM' 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', 'IM', 'Information Management' and the 'enterprise resource management' are developed and implemented using the 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 Branch Manager to modify the daily currencies rate between EURO and USD.
- (2) The EOS Interface is used to transmit the request job, restrict any operation on the previous rate from any teller and create the process.
- (3) The EPM is used 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: (1) to preserve the correct rate value from the Information Management federate and (2) to request from the Resource Management federate to assign the specified external resources (Rate Application, Printer and Teller) 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 used 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 modifies the exchange rate on the Rate Application, prints out the operation receipt and notifies the teller to use the new exchange rate.

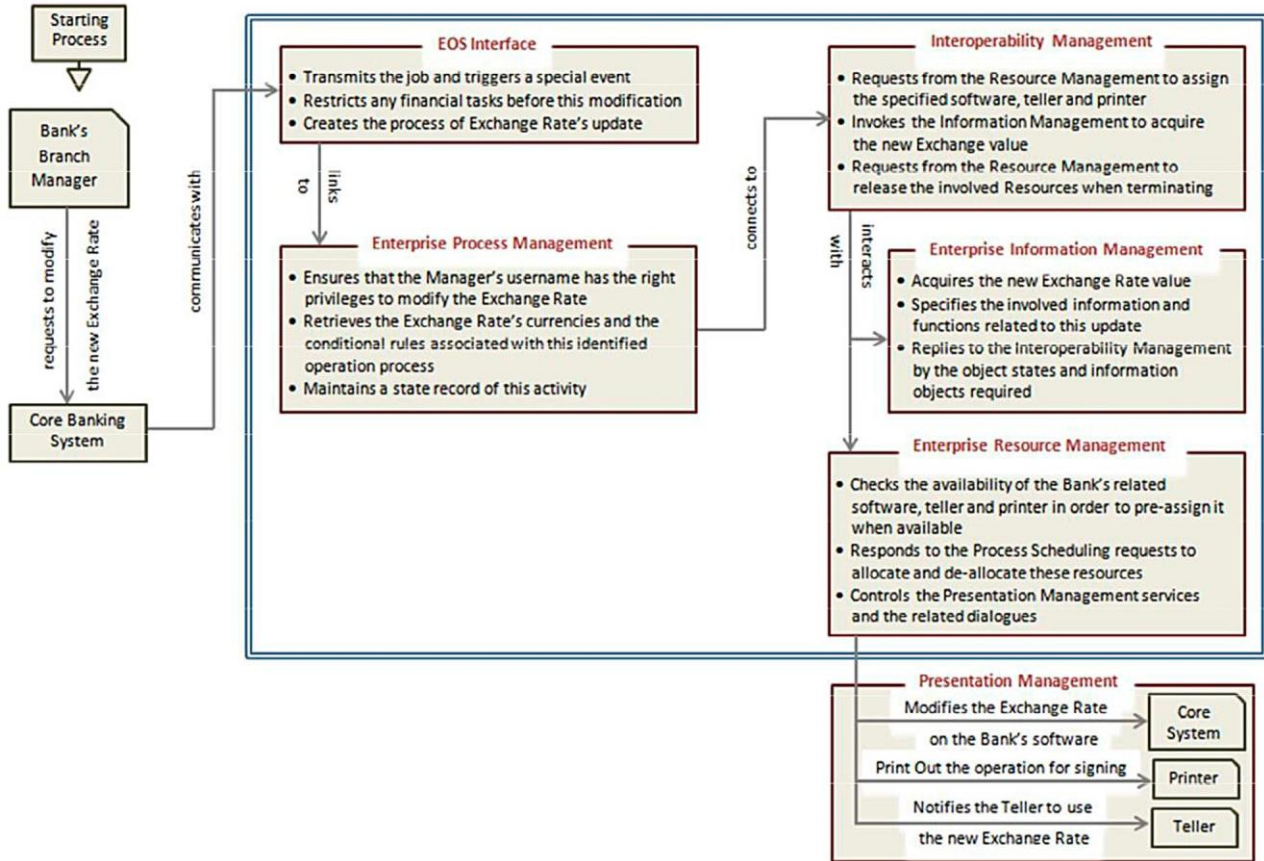


Figure 4. Conceptual and technical architecture of the bank exchange rate update.

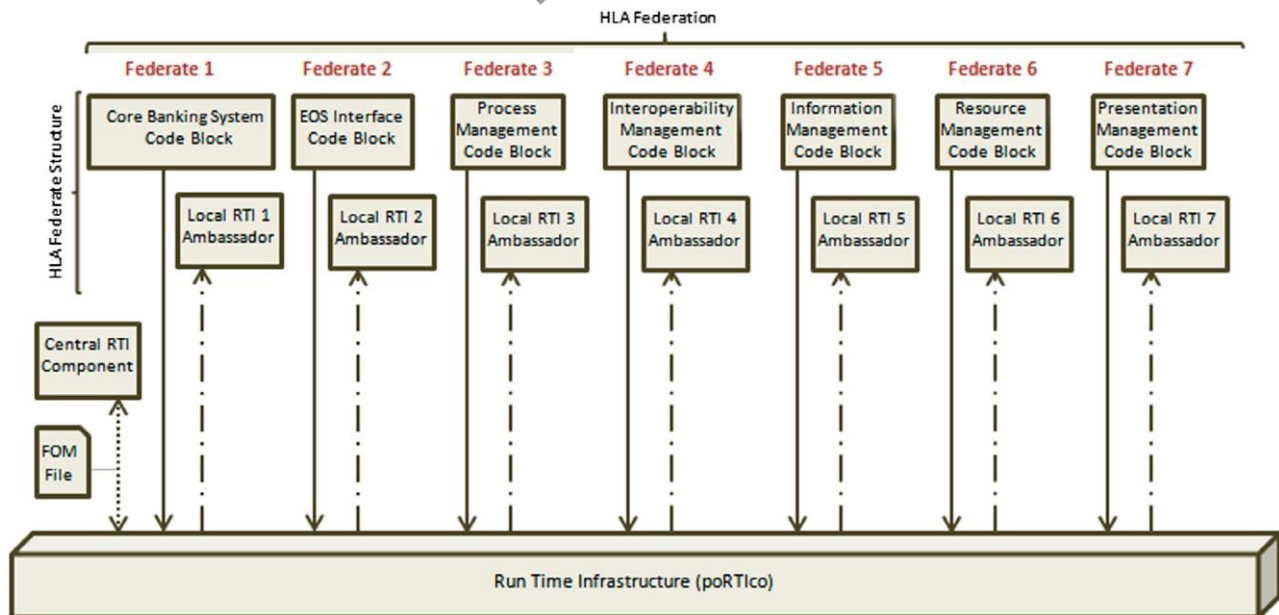


Figure 5. Implementation architecture of the bank exchange rate update.

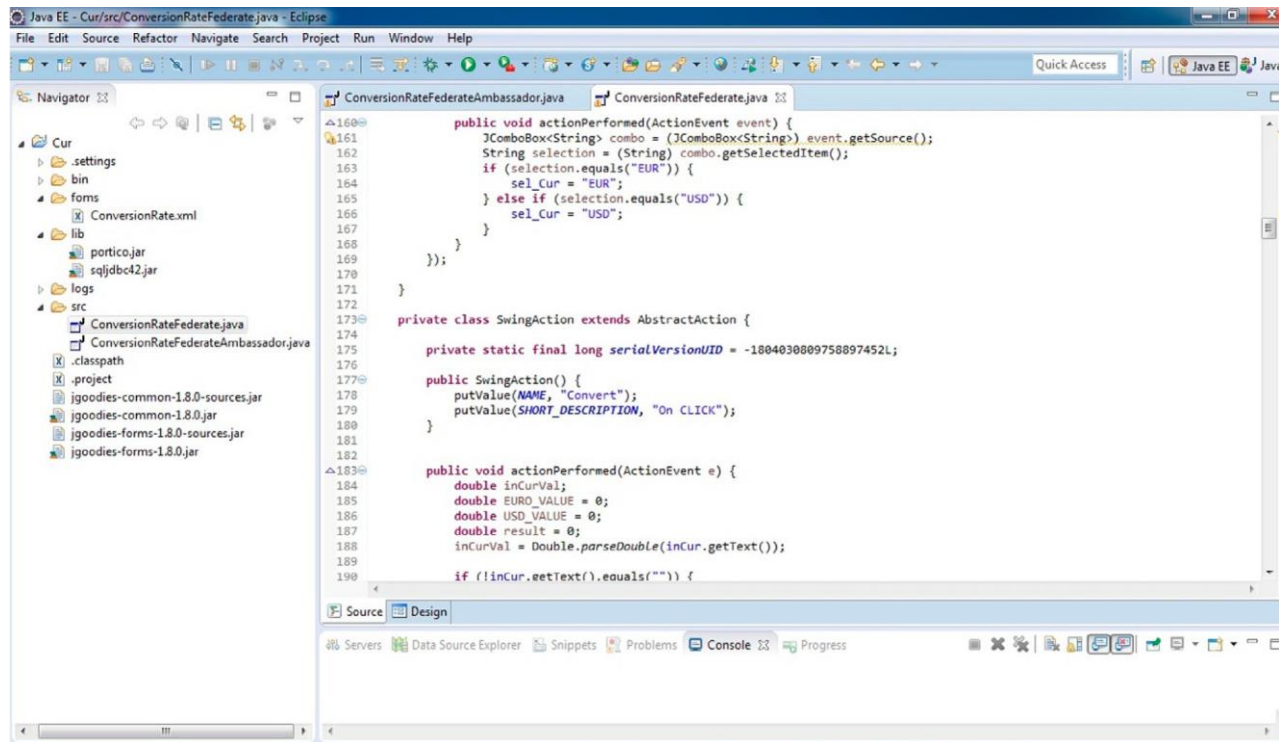


Figure 6. The EPM federate integrated with portico.

Concerning technical development and implementation, Figure 6 shows the EPM federate that is 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' 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.

According to the HLA specification, 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 has been successfully implemented over several heterogeneous environments 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.

6.2 Manufacturing sector – case example

This example illustrates how an EOS behaves in a manufacturing enterprise to monitor and control enterprise operations. It focuses on business operations performed by the EOS without detailing its internal and technical functioning.

The studied enterprise is a SME and works on 'Production to order'. The main products are 'Modular shelves' with standard and customised (with new design) options. The resources of the company are of the three types that one can encounter in most of the companies (computing, machining and human):

- Commercial service: Excel software (PC)
- Production Management: MRPEasy software (PC)
- Design service: DeltaCAD software
- Delivery: DISPATCH! software
- Machining: NC machine

- Assemble: Human operators
- Painting: Robots (3 units)

The main problems encountered in the enterprise are: (i) impossibility to know in real time what is happening in the workshop; manufacturing orders are often late due to the unavailability of resources; (ii) too much dead-time between operations due to inadequate planning and synchronisation of these operations and (iii) some enterprise software applications are not interoperable (e.g. MRPEasy, DeltaCAD and DISPATCH!), it is necessary to arrange the structure of some data tables and perform the translation of some terms used in different software.

The motivation of the enterprise to implement an EOS is to be able to more efficiently monitor and control its resources and the daily operations carried out on these resources. More specifically, the aims of the case example are to:

- Show that the resources can work together and be coordinated by the EOS, in particular:
 - Resources are monitored in real time (available, occupied, etc.)
 - Resources assignment is done in real time
- Show that heterogeneous applications can interoperate through the EOS, in particular:
 - Semantic interoperability (data meaning)
 - Syntax interoperability (data structure)

The commercial service receives customer orders and registers them using a simple Excel software. Orders requesting standard products will be directly sent to Production Management to plan their production while those requiring a customization will be first processed by Design Service and then sent to Production Management. Production Management will issue purchasing orders (send to external supplier) and manufacturing orders for production to NC machine and assembly line. Purchased raw materials will be first stored in the inventory, then moved to the machining workstation (NC machine) to manufacture shelves components. Then, the assembly operations are done by human operators to obtain shelves. The assembled shelves are painted using a robot (only one robot is needed at a time for one order).

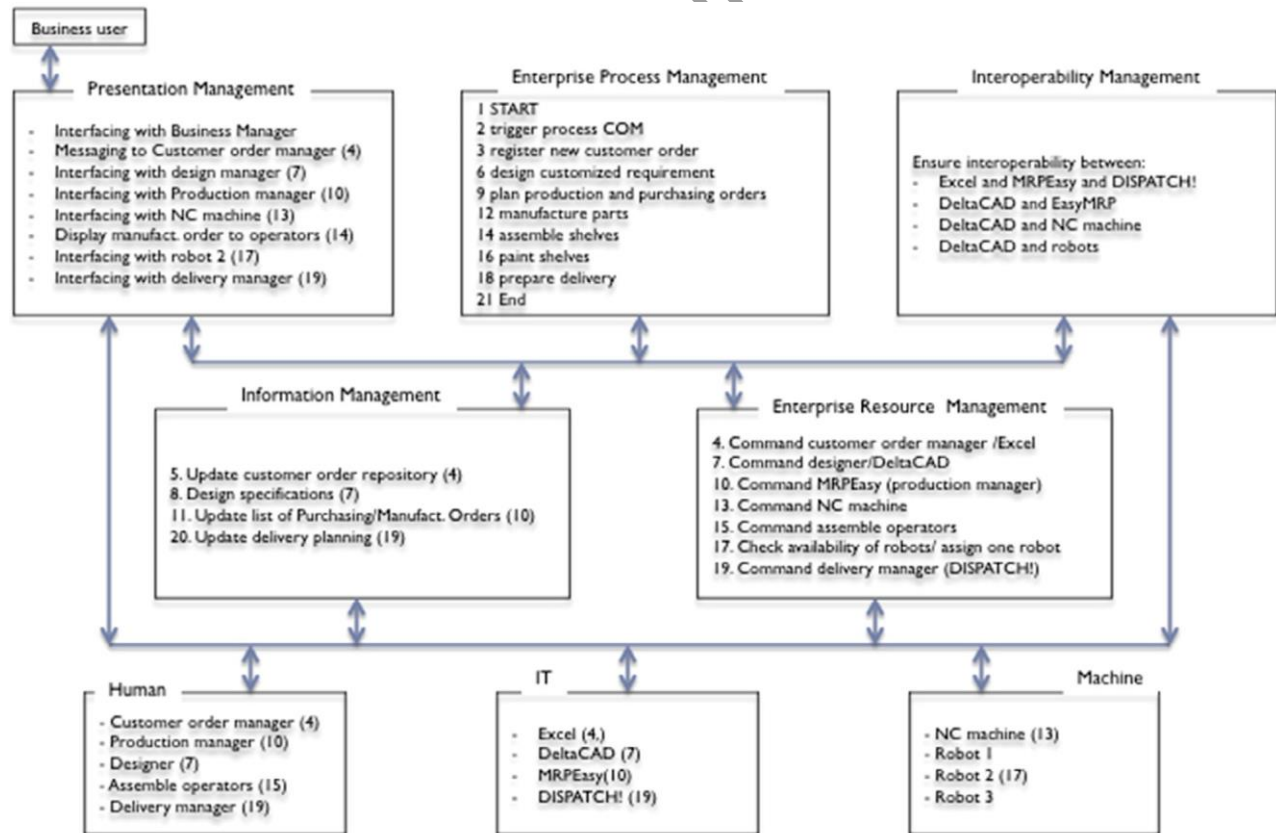


Figure 7. Manufacturing case scenario.

Then the finished products are delivered to customers according to a pre-defined delivery plan. Figure 7 illustrates how the EOS deals with one particular customer order that necessitates a redesign for a customised shelf.

The purpose of the presented scenario is to illustrate how the envisioned EOS can coordinate and command the needed operations to produce required products. The set of operations are numbered in sequence as shown in Figure 7. The resource needed and assigned by the EOS to perform each requested operation (with the corresponding operation number in bracket) is also indicated.

- (1) Arrival of a new customer order with a new design requirement. The business manager decides to validate the order and asks to start the operations (EPM).
- (2) EPM triggers the COM (Customised Order Management) process (it is a pre-defined process stored in the business process repository)
- (3) EPM interprets the first operation 'register new customer order' and sends a request to ERM
- (4) ERM informs the customer order manager via the 'PM' interface about the request
- (5) The Excel file of customer order list is updated (IM)
- (6) EPM interprets the next operation of the COM process: 'design product' to meet the customised requirements
- (7) ERM checks the availability of required resources and triggers Designer/DeltaCAD application
- (8) Design specifications are saved in a set of files with the help of Information management (IM)
- (9) EPM interprets the next operation: 'plan purchasing and manufacturing orders'
- (10) ERM checks the availability and triggers Production manager/MRPEasy application
- (11) Purchasing and Manufacturing Order lists are updated with the help of IM. Purchasing order is sent to external suppliers
- (12) EPM interprets the next operation 'manufacture parts'
- (13) ERM checks the availability of required resource and triggers 'NC machine' via PM. The 'NC machine' receives the technical specifications of parts via IM to manufacture the needed parts
- (14) EPM interprets the next operation: 'Assemble shelves'.
- (15) ERM checks the availability of human operators and sends a 'start to work' command via the PM interface. Operators receive technical specifications of shelves via IM to perform the assemble operations
- (16) EPM interprets the next operation: 'Paint shelves'
- (17) ERM checks the availability of 'robots' and assign one available robot (robot 2) to the task
- (18) EPM interprets the last operation: 'prepare delivery'
- (19) ERM informs the delivery manager and ask him to plan delivery using DISPATCH!
- (20) The delivery plan is updated (IM)
- (21) End of the process

It must be noted that the scenario described above has hidden some technical details on information exchanges between EOS functions. For example:

- at the end of each operation, a feedback information is sent to the EOS by the resource concerned to de-allocate the resource and trigger the next operation defined in the process;
- some interoperability engineering activities are required before executing some operations. A set of Interoperability Utility services in IM module is responsible for making necessary mappings and harmonisation between the software applications.

7. Conclusion and perspectives

The aim of the paper was to tentatively present requirements and architectures for developing an EOS for the emerging generation of new enterprise systems as envisioned in Industry 4.0. A detailed state-of-the-art analysis has been performed to assess existing relevant works. It has been concluded that none of the existing approaches/ initiatives was previously developed as an enterprise operating system per se. Although they all contribute to some extents to support enterprise operation management, they failed to act as a system-wide interface between business managers who monitor and control enterprise systems as a whole, and enterprise resources performing local daily enterprise operations. The proposed EOS tends to reconcile two different but complementary approaches/initiatives for enterprise management and control that exist in the market: IT platforms/infrastructure and ERP-based application packages as assessed in the state-of-the-art.

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 Operating System for production enterprises. EOS will mainly provide an alternative for Small- and Medium-sized Enterprises, support establishing enterprise interoperability in a heterogeneous environment based on the federated approach and facilitate re-use of models and re-engineering sub-systems based on models. 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 applications compatible to EOS. The other one is the ‘Interoperability’ service that allows other heterogeneous non-EOS compatible applications to run on EOS.

The work presented in the paper is still at the prototype level and is aimed at concept-proving. Future work planned is to refine and detail the technical architecture for EOS including all processes and related functionalities on the one hand and, on the other hand, to develop the EOS implementation architecture and a generic prototype to test the EOS against two use cases in both manufacturing and service sectors. It is also necessary to foster some preliminary discussion to build an international consensus in relevant standardisation bodies dealing with enterprise engineering, integration and interoperability such as CEN TC310 WG1 and ISO TC184 SC5/WG1.

References

- AIIM. 2000. *Enterprise Applications – Adoption of E-Business and Document Technologies*. AIIM BookStore. https://www.techstreet.com/direct/ExSum_NA.pdf.
- AMICE. 1993. *CIMOSA– Open System Architecture for CIM, 2nd and Extended Revision*, Esprit Consortium AMICE. Berlin: Springer.
- Bryan, K., L. C. Dipippo, and V. Fay-Wolfe. 2005. *Integrated CORBA Scheduling and Resource Management for Distributed Real-Time Embedded Systems (IEEE Xplore, March 2005)*.
- Burnson, F. 2015. *Enterprise Resource Planning Software – Buyer Report*. Stamford, CT: Software Advice™.
- CEITON. 2014. “Front-End and Back-End EAI.” *CEITON Technologies*, Wikipedia. Accessed May 2014. https://en.wikipedia.org/wiki/Enterprise_application_integration.
- CEN. 1999. “prENV 13550 Advanced Manufacturing Technology–Systems Architecture–Enterprise Model Execution and Integration Services.” Secrétariat du CEN TC 310 (dot. CEN TC 310 N 706).
- Chang, J. F. 2005. *Business Process Management Systems, Strategy and Implementation*. Boca Raton, FL: CRC Press.
- Chappell, D. A. 2004. *Enterprise Service Bus*. O’Reilly Media. ISBN: 0596006756. ACM Digital Library.
- Chen, D. 2005. “Enterprise-Control System Integration – An International Standard.” *International Journal of Production Research*, Taylor & Francis 43 (20): 4335–4357.
- Chen, D., G. Doumeingts, and F. Vernadat. 2008. “Architectures for Enterprise Integration and Interoperability: Past, Present and Future.” *Computers in Industry* 59: 647–659.
- Chen, D., J. R. Youssef, and G. Zacharewicz. 2015. “Towards an Enterprise Operating System – Requirements for Standardisation, *Proceedings of IWEI 2015 (IWEI Workshops 2015)*, May 2015.
- Fagg, G. E., K. S. London, and J. J. Dongarra. 1996. *Taskers and General Resource Managers, Third European PVM Users’ Group Meeting, EuroPVM’ 96*. Lecture Notes in Computer Science, vol 1156. Berlin, Heidelberg: Springer.
- Flurry, G., and K. Clark. 2011. *Enterprise Service Bus*. © Copyright IBM Corporation 2011. *TechJournal*, May 2011.
- Gable, J. 2002. “Enterprise Application Integration.” *Information Management Journal* (Magazine Article). Online research. April 2002.
- Gray, J., and A. Reuter. 1993. *Transaction Processing: Concepts and Techniques*. Vol. 1, 80–93. San Francisco, CA: Morgan Kaufmann Publishers (An Imprint Elsevier).
- Henning, M. 2006. Response to ‘the Rise and Fall of CORBA’ (ACM 2006 Article, June 2006. <https://queue.acm.org/detail.cfm?id=1142044>).
- Hollinsworth, D. 1994. “The Workflow Reference Model”. *Journal of Software Engineering and Applications*. Document No. TC00-1003, No. 1.1, 1995.
- IEEE 1516. 2010. “Standard for Modelling and Simulation High Level Architecture”. *Framework and Rules. IEEE International Conference on 2011*, 1–9. New York: IEEE Xplore – Digital Library. Online ISBN: 978-0-7381-6251-5.
- ISO/IEC. 2009. *ISO/IEC 10746:2009, Information Technology – Open Distributed Processing – Reference Model: Architecture*. Geneva: International Standards Organization.
- Jin, X. 2009. “Research on the Model of Enterprise Application Integration with Web Services”. *Proceedings of the 3rd WSEAS International Conference on Computer Engineering and Applications (CEA’09)*, Beijing, China. ISBN: 978-960-474-41-3.

- Knight, P., A. Corder, R. Liedel, J. Giddens, R. Drake, C. Jenkins, and P. Agarwal. 2002. "Evaluation of Run Time Infrastructure (RTI) Implementations." Huntsville Simulation Conference, Huntsville, AL, USA.
- Konstantas, D., J. P. Bourrières, M. Léonard, and N. Boudjlida. 2006. "Interoperability of Enterprise Software and Applications." *INTEROP-ESA International Conference*, Geneva, Switzerland.
- Kosanke, K. 1995. "CIMOSA – Overview and Status." *Computers in Industry* 27 (1995): 101–109.
- Leadbeater, P. F. 1999. "Systems Architecture." *Enterprise Model Execution and Integration*, CEN/TC310, ENV13550 40 (2–3): 305–310.
- Linthicum, D. S. 1999. *Enterprise Application Integration*, Vol. 20, 14–126. Addison-Wesley Information Technology Series.
- May, J. M. 2001. *Parallel I/O for High Performance Computing*. Boston, MA: ACM Digital Library.
- Moon, J., and S. Lee. 2006. "Design and Implementation of a Resource Management System Using on-Demand Software Streaming on Distributed Computing Environment." *Proceedings of the 6th International Conference on Computational Science - Volume Part I*, Berlin, Germany.
- OMG. 2012. *Common Object Request Broker Architecture (CORBA), Version 3.3*, Object Management Group. <https://www.omg.org/spec/CORBA/3.3>.
- Omicini, A., P. Petta, and J. Pitt. 2003. *Engineering Societies in the Agents World 4th International Workshop London, UK, 2013*.
- poRTIco RTI. 2009. www.porticoproject.org.
- Rademakers, T., and J. Dirksen. 2008. "Open-Source ESBs in Action" (Manning Publications, September 2008. <https://www.manning.com/books/open-source-esbs-in-action>).
- Rasta, K. 2013. *Data Quality-Based Resource Management in Enterprise Service Bus*. Kristiansand: University of Agder.
- Saha, D., A. Mukherjee, and S. Bandyopadhyay. 2011. *Networking Infrastructure for Pervasive Computing, Enabling Technologies*. Boston, MA: Springer/Kluwer Academic Publishers.
- Sethi, A. S., Y. Raynaud, and F. Faure-Vincent. 1995. "Integrated Network Management IV." *Proceedings of the Fourth International Symposium on Integrated Network Management*, Springer US.
- Shorter, D. N. 1997. *Requirements for Model Execution and Integration Services CEN/TC310 WG1*. Brussels: European Committee for Standardization.
- Turner, K. J. 2012. *Advances in Home Care Technologies Results of the MATCH Project*. Amsterdam: IOS Press.
- Vallejo, C., D. Romero, and A. Molina. 2012. "Enterprise Integration Engineering Reference Framework and Toolbox." *International Journal of Production Research*, Taylor & Francis 50 (6): 1489–1511.
- Vollmer, K. 2011. "The Forrester Wave™: Enterprise Service Bus for Application Development & Delivery Professionals." *Forrester Research Q2*.
- Wainer, J., R. S. Filho, and E. R. M. Madeira. 2000. *CORBA Based Architecture for Large Scale Workflow*. Tokyo: Institute of Computing.
- Weichhart, G., A. Molina, D. Chen, L. E. Whitman, and F. Vernadat. 2016. "Challenges and Current Developments for Sensing, Smart and Sustainable Enterprise Systems." *Computers in Industry* 79: 34–46.
- WfMC. 1999. *The Workflow Management Coalition Specification, Workflow Management Coalition, Terminology & Glossary*. WfMC-TC-1002. March 99. Winchester: Baidu.
- Wyszkowski, P. 2011. *ESB Application for Effective Synchronization of Large Volume Measurements Data*. Kraków: University of Science and Technology, ResearchGate.
- Xu, L. D. 2014. *Enterprise Integration and Information Architecture, A Systems Perspective*, Book Depository. Toronto: Toronto Public Library.
- Youssef, J. R., G. Zacharewicz, and D. Chen. 2016. "Developing an Enterprise Operating System (EOS) – Requirements and Architecture." *WETICE IEEE conference*. Paris.
- Youssef, J. R., G. Zacharewicz, D. Chen, and Z. Tu. 2016. "Developing an Enterprise Operating System (EOS) with the Federated Interoperability Approach." *I3M Conference*. Chypre.
- Zaraté, P. 2013. *Tools For Collaborative Decision-Making*. Wiley Online Library. Accessed February 2013. <https://onlinelibrary.wiley.com/doi/10.1002/9781118574690.ch2/summary>.
- Zurawski, R. 2004. *The Industrial Information Technology Handbook*. Boca Raton, FL: CRC Press.