

## A federated approach to develop enterprise interoperability

Zhiying Tu, Gregory Zacharewicz, David Chen

► **To cite this version:**

Zhiying Tu, Gregory Zacharewicz, David Chen. A federated approach to develop enterprise interoperability. *Journal of Intelligent Manufacturing*, Springer Verlag (Germany), 2016, 27 (1), pp.11-31. <http://link.springer.com/article/10.1007%2Fs10845-013-0868-1> . 10.1007/s10845-013-0868-1 . hal-00937638

**HAL Id: hal-00937638**

**<https://hal.archives-ouvertes.fr/hal-00937638>**

Submitted on 26 Jan 2016

**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.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/260339930>

# A federated approach to develop enterprise interoperability

ARTICLE *in* JOURNAL OF INTELLIGENT MANUFACTURING · FEBRUARY 2016

Impact Factor: 1.73 · DOI: 10.1007/s10845-013-0868-1

---

CITATION

1

---

READS

55

### 3 AUTHORS:



[Zhiying Tu](#)

Harbin Institute of Technology

7 PUBLICATIONS 14 CITATIONS

[SEE PROFILE](#)



[Gregory Zacharewicz](#)

University of Bordeaux

86 PUBLICATIONS 190 CITATIONS

[SEE PROFILE](#)



[David Chen](#)

Shih Hsin University

107 PUBLICATIONS 1,122 CITATIONS

[SEE PROFILE](#)

---

# A Federated Approach to develop Enterprise Interoperability

Zhiying Tu<sup>(a)</sup>, Gregory Zacharewicz<sup>(b)</sup> David Chen<sup>(c)</sup>

<sup>(a)(b)(c)</sup> IMS-LAPS / GRAI, Université de Bordeaux – CNRS, 351, Cours de la Libération,  
33405, TALENCE cedex, France

<sup>(a)</sup> zhiying.tu@ims-bordeaux.fr, <sup>(b)</sup> gregory.zacharewicz@ims-bordeaux.fr, <sup>(c)</sup> david.chen@ims-bordeaux.fr,

## Abstract

Interoperability is one of the requisite features for existing enterprises in the increasing competitive and complex global market. In the last decade, enterprise interoperability has been developed and prescribed by various kinds of frameworks, methods, and techniques. However interoperability development is still not mature enough to become a science. Meanwhile, it keeps evolving according to different business requirement and market environment. Nowadays, networked environment causes unpredictable dynamical situations, thus sustainable interoperability becomes a new research dimension in the interoperability of enterprise systems and applications domain. In the sustainable interoperability, enterprise interoperability dynamics is one of the focal topics. This dynamic approach also called federated is originated from Enterprise Interoperability Framework of INTEROP NoE, which aims to establish interoperability on the fly. This paper presents current state on federated approaches to develop enterprise interoperability dynamics. Based on this study, a novel Federated interoperability approach is proposed. It aims at bridging the gap from interoperability concepts to the implementation of interoperable enterprise Information Systems development. This approach reuses distributed simulation interoperability concepts to facilitate and coordinate the communication between heterogeneous distributed information systems of the enterprises. This simulation part has been implemented into a software platform. This platform is compliant with the latest version of the High Level Architecture (HLA) that is a distributed communication standard. This approach also proposes a development lifecycle that intends to reuse existing information systems without recoding them but by adapting them to the new requirements of interoperability dynamics.

Key words: Enterprise Interoperability; Dynamic; Federated approach, HLA, MDA, Model Reverse

## 1. Introduction

Nowadays, enterprise collaboration becomes more and more important because of globalized economic context. In this context, an enterprise often interoperates at the same time with many different heterogeneous partners having different technologies, semantics, methods of work and organizations. Today, most of existing interoperability solutions proposes integrated or unified approaches which are not satisfactory to dynamic

---

networking enterprises. It has been considered that the federated approach provides the best perspective to develop full enterprise interoperability in the future.

The objective of the paper is to propose a federated approach for developing enterprise interoperability. In this approach, cooperating parties must accommodate and adjust “on the fly” to establish interoperability. “On-the-fly” means all the models mapping need to be done dynamically through “negotiation”. In other words, federated approach has no common predefined format for all models and needs dynamic adjustment and accommodation. This paper will present a solution to achieve federated approach and demonstrate the feasibility of the proposed solutions with some concrete examples and a case study.

This paper is structured as follows. Chapter 2 will introduce the enterprise interoperability (EI) science base, including EI definitions, EI dimensions, EI research tendency and challenges, and existing projects in EI Dynamics. Chapter 3 will present a State-of-the-art survey on existing architectures, techniques and methodologies that are relevant to the development of federated enterprise interoperability, including model driven technologies, and simulation and application distribution frameworks/technologies. Chapter 4 will elaborate a harmonized and reversible development framework and methodology for developing federated enterprise interoperability. Chapter 5 will illustrate a case study of the proposed federated approach. Finally, chapter 6 will summarize the achievement of this federated approach, and point out the open issues.

## **2. Enterprise Interoperability Science Base**

This chapter will introduce the basic concept of enterprise interoperability, and elaborate the tendency and challenges of the current enterprise interoperability research. Finally, some existing projects that aim at developing enterprise interoperability dynamics will be introduced.

### **2.1. Enterprise Interoperability Definitions**

Interoperability is a key feature for enterprises in today’s competitive environment. Generally, “Inter-operate” implies that one system performs an operation on behalf of (or for) another system. However, Interoperability means different things to different people, so there are various kinds of definitions as follows:

- Interoperability is the ability of a system to use the parts of another system – definition in Webster.
- The definition of Interoperability in IEEE is “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” (IEEE 1990).
- European Interoperability Framework defines interoperability as “is the ability of disparate and diverse organizations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organizations, through the business processes they support, by means of the exchange of data between their respective ICT systems.” (EIF 2011).

These definitions describe interoperability from different aspects. Some definitions describe the interoperable system’s/organization’s behavior. Some others emphasize the information interoperability. Some definitions consider software application interoperability. While, the definition from IDEAS focuses enterprise

---

interoperability on business processes interoperability, not only information interoperability. To summarize those definitions, Enterprise Interoperability is the ability to (1) communicate and exchange information; (2) use the information exchanged; (3) access to functionality of a third system (Chen 2009).

However, in the last few years, some researches considered that those definitions need to be extended to cover the additional interoperability issues in the enterprises, and a broader, more comprehensive definition is needed. As a result, some new definitions of Enterprise Interoperability were given in different projects.

- Enterprise Interoperability Research Roadmap (EIRR) define Enterprise Interoperability as “a field of activity with the aim to improve the manner in which enterprises, by means of Information and Communications Technologies (ICT), interoperate with other enterprises, organizations, or with other business units of the same enterprise, in order to conduct their business. This enables enterprises to, for instance, build partnerships, deliver new products and services, and/or become more cost efficient” (Charalabidis 2008).
- European Interoperability Framework defines interoperability as “the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable the sharing of information and knowledge” (IDABC 2008). It also indicates “Interoperability is the ability of disparate and diverse organizations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organizations via the business processes they support, by means of the exchange of data between their respective information and communication technology (ICT) systems” (EIF 2011).

These definitions involve interoperability between organizational units and business processes and units either within distributed enterprises or within an enterprise network. In a word, Enterprise Interoperability is perceived as a capacity of two or more enterprises, including all the systems within their boundaries and the external systems that they utilize or are affected by, in order to cooperate seamlessly, in an automated manner, in depth of time for a common objective (ENSEMBLE 2011) ([Gonçalves 2012](#)).

## 2.2. Enterprise Interoperability dimensions

To better understand the Enterprise interoperability concept, to define and position the approach proposed in this paper, it is necessary to study various dimensions of enterprise interoperability. Those dimensions representing problems, issues and concerns of Enterprise interoperability research and development are usually structured and represented in enterprise interoperability frameworks. Figure 1 shows the INTEROP Enterprise interoperability Framework (now CEN/ISO 11354 standard) (Chen 2008) with its three main dimensions:

[Figure 1 Enterprise Interoperability Framework]

- “*Interoperability concerns*” which identifies the area of interoperation that may take place at various levels of the enterprise (data, service, process, business), i.e. the level at which the interoperation occurs.
- “*Interoperability barriers*” which identifies various obstacles to interoperability in three categories (conceptual, technological, and organizational), i.e. the type of obstacle to interoperability.

- 
- “*Interoperability approaches*” which represents the different ways in which barriers can be removed (integrated, unified, and federated).

The objective of this framework is to tackle interoperability problems through the identification of barriers that prevent interoperability to occur. The first two dimensions, Interoperability concerns and Interoperability barriers, constitute the problem space of enterprise interoperability. After identifying the problem space, the “*Interoperability approaches*” proposes the solution to this problem space. The definitions of the integrated, unified, and federated approaches are as follows:

- *Integrated Approach* means that there exists a common format for all models. Diverse models are built and interpreted using/against the common template. This format must be as detailed as the models themselves. The common format is not necessarily an international standard but must be agreed by all parties to elaborate models and build systems.
- *Unified Approach* means there is a common format but it only exists at meta-level. This format is not an executable entity as it is the case in integrated approach. Instead it provides a mean for semantic equivalence to allow mapping between models and applications. Using the meta-model a translation between the constituent models is possible even though they might encounter loss of some semantics or information.
- *Federate Approach* aims to establish the interoperability ‘on the fly’, which means the interoperability accommodating and adjusting should not impose the existing models, languages and methods of work as the common format.

The integrated and unified approaches have been well implemented, but the federate approach is still an on-going research and it is the concern of this paper. In the enterprise interoperability roadmap published by the European Commission (Charalabidis 2008), developing federated approach for interoperability is considered as one of the research challenges for the years to come.

## **2.3. Enterprise Interoperability Research Tendency and Challenges**

Nowadays, under the globalised economic context, the markets are becoming more and more competitive and complex. The complex markets require the enterprises to adapt in the dynamic and changing environment. That means Enterprise Interoperability should become more and more related with the dynamic nature of future business requirements, both for the single enterprise and ecosystems.

As mentioned in the Enterprise Interoperability definition of EISB (Enterprise Interoperability Science Base) (ENSEMBLE 2011) ([Gonçalves 2012](#)), in order to achieve the common goal and realize the enterprise interoperability, the enterprises need to cooperate seamlessly, in an automated manner, in depth of time. This means that they need a very efficient, dynamic, sustainable and seamless approach/solution, as assumed in Enterprise Interoperability Dynamics. The Enterprise Interoperability Dynamics is the aspiration that the enterprises can be networked fluently, efficiently, dynamically, intelligently and with the lowest cost. In that

---

case, all the collaboration operations can be established “on-the-fly”. Such as, enterprises do not need to think about reconstructing their legacy systems or building up an integrated platform to support cooperation. A potential participant can detect this collaborative sphere and access it easily, and the existing participants can detect the new partner dynamically and evolve themselves to adapt the new environment.

According to the tendency mentioned above, some of main challenges towards dynamic enterprise interoperability are considered as follows:

- Dynamic market and economic context require an enterprise capable of interoperating simultaneously with multiple heterogeneous partners. This means that an enterprise must be able to adjust and adapt their systems constantly and without delay.
- To adapt and accommodate dynamically to potential interoperability partners, it is necessary to perform ‘on-the-fly’ needed changes and mapping of systems connected to partners. Consequently the ability to quickly reengineering enterprise systems is an important challenge.
- Prior to any reengineering attempt, another challenge is to be able to model and automatically collect relevant information and data on the legacy systems and software applications already implemented in the enterprise and concerned by the interoperation.
- To dynamically establish interoperability, it is necessary to reduce complexity in Enterprise Interoperability. How to use interoperability services as “plug-and-play” mechanisms independently of the Enterprise Interoperability level for which they are designed (higher levels such as business, or lower ones such as technical applications) is another challenge to consider in this research.

## **2.4. Existing projects in Enterprise Interoperability Dynamics**

Recently, some projects continue the research and development in enterprise interoperability domain, focusing more specifically on dynamic enterprise interoperability approach. We can mention among others:

- ABILITIES (Application Bus for Interoperability In enlarged Europe SMEs) project is a FInES (Future Internet Enterprise Systems) Cluster in FP6. Its objective is to study, design and develop a federated architecture implemented by a set of intelligent and adaptive UBL active messages (an Application Bus for EAI - Enterprise application integration) and basic interoperability services, aiming at supporting SMEs EAI in e-commerce contexts, specifically in less developed Countries and less RTD intensive industrial sectors (ABILITIES 2008).
- COIN (Enterprise Collaboration & Interoperability) is one of the FInES Cluster FP7 Projects (COIN 2011). Its objective is to study, design and develop an open, self-adaptive, generic ICT integrated solution to support the 2020 vision, starting from notable existing research results in the field of Enterprise Interoperability and Enterprise Collaboration. COIN project believes that Enterprise Interoperability and Enterprise Collaboration (two different concepts) can be interdependently and simultaneously presented in every networked enterprise.
- NisB (The Network is the Business) project is one of the FInES Cluster FP7 Projects (NisB 2010). It aims at providing ICT support for value networks of SMEs, namely hierarchical supply chains or dynamic business ecosystems, thereby rendering them the primary facilitators of innovative networked businesses. NisB allows small and medium businesses to easily and affordably connect, align, exchange data and complete transactions with peers they have little common business language with.

- 
- GENESIS (Enterprise application interoperability via Internet-Integration for SMEs, Governmental Organisations and Intermediaries in the new European Union) project is one of the FP6-IST projects (GENESIS 2006). It aims at enlarging the scope and objectives of interoperability of infrastructures to full scale business integration platforms. It developed a prototype system to enable the typical, usually small and medium, East-European enterprises to connect their software systems with the systems of collaborating enterprises, banking/social insurance institutions and governmental bodies, with respect to the evolving legal and regulatory status. Therefore, they can conduct their Business transactions over Internet.
  - FUSION (Business process FUSION based on Semantically-enabled Service-Oriented Business Applications) project is one of the FP6-IST projects (GENESIS 2008). It aims at promoting efficient business collaboration and interconnection between enterprises (including SMEs). It developed an innovative approach, methodology and integration mechanism for the semantic integration of a heterogeneous set of business applications, platforms and languages within SMEs.

These projects have achieved good results in Enterprise Interoperability Dynamics, but still have certain limitations. ABILITIES project has some comment format for information exchange, which does not fully comply with federated approach of the Enterprise Interoperability Framework. The present COIN solutions focus on data concern more. NisB project has adopted the federated approach at the knowledge level and not at the service level. GENESIS project develop a prototype system that allows users to collaborate via their existing software application or GENESIS web application. The messages/documents are dispatched by GENESIS server. It is an interoperability platform, but not federated one. FUSION project raised the integration and interoperability from “data” level to “process” level, but it still emphasized “EAI (Enterprise Applications Integration)”.

## 3. Technical recall

This chapter will present the state-of-the-art on models, architectures, techniques and methodologies which are helpful for the development of federated enterprise interoperability.

### 3.1. Model driven Architecture

The life cycle development methodology studied firstly is Model Driven Architecture (MDA). This methodology has been defined and adopted by the Object Management Group (OMG) in 2001, (updated in OMG 2003). It is designed to promote the use of models and their transformations to consider and implement different systems. It is based on an architecture defining four levels, which go from general considerations (conceptual level) to specific ones (implementation level).

- *CIM Level* (Computation Independent Model): focusing on the whole system and its environment. It is also named “domain model”. It describes all work field models (functional, organisational, decisional, process, etc.) of the system with a vision independent from implementation.



- 
- *PIM Level* (Platform Independent Model): modelling the sub-set of the system that will be implemented, but does not show the details of its use of its platform. It might consist of enterprise, information and computational viewpoint specifications.
  - *PSM Level* (Platform Specific Model): taking into account the specificities related to the development platform.
  - Coding Level: last level, consisting in coding enterprises applications (ESA: Enterprise Software Application).

To complete this description, a Platform Description Model used for the transformation between PIM level and PSM level is added to these four kinds of models corresponding to the four abstraction levels.

## 3.2. Model Driven Interoperability framework

The approach "Model Driven Interoperability" (MDI) considers interoperability problems at the enterprise model level instead of only at the coding level.

MDI concepts were developed in the Task Group 2 (TG2) of INTEROP-NoE project by defining an approach inspired by the OMG MDA concepts (Bourey 2007). The goal of MDI is to tackle the interoperability problems at each abstraction level defined in MDA and to use model transformation technique to link both vertically the different levels of the MDA abstraction and horizontally the corresponding models of the systems to interoperate. The main goal of MDI, based on model transformation, is to allow a complete follow-up from the expression of requirements to the coding of solutions and also to provide a greater flexibility thanks to the automation of these transformations.

In the context of TG2, experimentations have been realized and in particular the feasibility study to transform GRAI Methodology (Doumeingts 2001) Models to UML models between CIM and PIM levels (Bourey 2007). These works are complemented by additional works realized in the context of ATHENA to define UML profiles to take into account also the Service Oriented Architectures (SOA) at the PIM level ([Gorka 2007](#)). These results have been complemented by results presented by ([Touzi 2007](#)) who has proposed an interoperability transformations method from BPMN to UML in the context of SOA.

Nevertheless, the soundness of the methodology has been demonstrated, but no full industrial scale validation has been yet achieved. Some projects have especially carried out the demonstration of these concepts in an industrial real world significant application. The different methodological propositions are tested and refined by focusing on models and their interoperability. They consist in particular of ways to improve the flexibility of the MDI transformation process and in obtaining dynamic interoperability in the context of the federated approach.

---

### 3.3. Model Reverse Engineering

MDA is well-known for promoting the use of models and their transformations to design and implement different information systems. After MDA became an important change in software development, OMG launched another research activity leading to what was later called Architecture Driven Modernization (ADM) (OMG 2010).

The basic idea proposed in the MDA approach is to translate from an abstract platform-independent model (PIM) expressed in UML into a more concrete platform-specific model (PSM) from which the code still needs to be generated (OMG 2003). Reversing the MDA lifecycle, ADM is discovering models from the coding level of legacy information system, such as UML models, Knowledge Discovery Meta-model (KDM) and Abstract Syntax Tree Meta-model (ASTM). KDM and ASTM are aimed to satisfy someone interested in discover more specific models from a legacy system (OMG 2010).

KDM is a meta-model for representing existing software assets and their associations, as well as relationships among the function models in the system. It also describes the operation environments. It can insure the interoperability among the existing systems, make the data exchange among different vendor tools easier. ASTM aims at enabling easy interchange of detailed software metadata between software development and software modernization tools, platforms, and metadata repositories in distributed heterogeneous environments (OMG 2011). It defines a specification for modelling elements to express abstract syntax trees (AST) in a representation that is sharable among multiple tools from different vendors.

### 3.4. High Level Architecture

The High Level Architecture (HLA) is a software architecture specification that defines how to create a global software execution composed of distributed simulations and software applications. This standard was originally introduced by the Defence Modelling and Simulation Office (DMSO) of the US Department Of Defence (DOD). The original goal was reuse and interoperability of military applications, simulations and sensors.

In HLA, every participating application is called “federate”. A federate interacts with other federates within a HLA federation, which is in fact a group of federates. The HLA set of definitions brought about the creation of the standard 1.3 in 1996, which Evolved to HLA 1516 in 2000 (IEEE 2000).

The interface specification of HLA describes how to communicate within the federation through the implementation of HLA specification: the Run Time Infrastructure (RTI). 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. The information exchanged in HLA is represented in the form of classical object class oriented programming. The two kinds of object exchanged in HLA are Object Class and Interaction Class. Object class contains object-oriented data shared in the federation that persists during the run time; Interaction class data are just

---

sent and received information between federates. These objects are implemented within XML format. More details on RTI services and information distributed in HLA are presented in (IEEE 2000). In order to benefit from the Web Services such as, the support for numerous newer and older languages and operating systems as well as the ease of deployment across wide area networks, HLA Evolved IEEE 1516<sup>TM</sup>-2010 was published in August, 2010 (IEEE 2010).

The FEDEP (Federation Development and Execution Process) describes a high-level framework for the development and execution of HLA federation. FEDEP uses the seven-step process to guide the development of the simulation system through phases of (1) requirements, (2) conceptual modelling, (3) design, (4) software development, (5) integration, (6) execution and (7) evaluation (IEEE 2003). It has been recently integrated into the more general DSEEP (Distributed Simulation Engineering and Execution Process) framework (IEEE 2011).

### 3.5. Web Services

Web Services has achieved a great success in the business domain, which stems from the good characteristics of the technology itself, is widely recognized by enterprises and business organizations and provides effective support for the open source community (Richardson 2007).

The data exchange via Web services is based on open standards such as HTTP and XML and is not associated with any particular vendor, operating system and programming language, which makes Web services platforms vendor neutral. Coarse-grained business functions can also be packaged for the Web service platform and can then be discovered by potential consumers.

At the same time, business-to-Web service is widely recognized. Microsoft, IBM and Sun and other leading manufacturers as well as Apache and other open source organizations support it. World Wide Web and other standardization organizations with active participation in the Web services technology provide for great maturity and popularity of the organization advantage.

## 4. Specification of the federated approach

This chapter presents a harmonized and reversible engineering framework and methodology for developing federated approach. This approach aims at developing a HLA based application to set up interoperability rapidly among existing enterprise information systems. The framework illustrated in figure 2 has three primary concepts that can be separately presented as follows.

*Harmonized* means that this framework is synthetic, which consists of several techniques. As the framework in figure 2 shows, we propose a new five steps development life cycle which aligns MDA and HLA FEDEP. MDA is easy to use and understand, and tightly bounded with Unified Modelling Language, Meta-Object Facility (MOF). It appears to be an appropriate solution to overcome the interoperability barriers, such as the

---

MDI framework mentioned in (Elvesæter 2007). HLA FEDEP is the standard for development and execution of HLA federation. It is quite similar to the waterfall development but with look-back test phase. MDA and HLA FEDEP can be easily aligned, because they have several similar steps. The HLA FEDEP & MDA alignment will be explained in section 4.1. This framework also uses web services to improve the flexibility and compatibility of the HLA. The Web Services allows potential external systems to discover the existing HLA Federation, and then connect to it. Section 4.3 will explain the reason and way of using web services. In addition, short-lived ontology method will be applied under this framework for information analysis. This method will be presented in section 4.4.

*Reversible* means that this framework uses model reverse engineering technique to discover part of the models from the legacy system. Model reverse engineering technique aims at avoiding rebuilding the complete legacy system for a new reuse. The objective is to accelerate the development and reduce the cost. As figure 2 illustrates, there are two kinds of dotted arrows, which have opposite directions to the five steps development life cycle. These two kinds of arrows represent two different scenarios of model reversal in this framework. Section 4.2 will present the method of using model reverse technique to rapidly develop HLA based interface for achieving federated enterprise interoperability.

*HLA* means that this framework dedicates to the development of HLA based application. The RTI used in this approach is an open source RTI, poRTIco (poRTIco 2009). The reason of choosing it is not only because of the software price, but also the objective of initiating a global open framework and receiving comments from contributors who can be interested in this idea. In addition, as mentioned earlier in Harmonized part, Web Services will be used to improve the limitation of the traditional HLA. Thus, the HLA approach proposed in this paper is based on the HLA evolved IEEE 1516<sup>TM</sup>-2010 standard. This approach will be presented in section 4.3.

[Figure 2 Harmonized and reversible development framework]

## 4.1. Harmonized Framework of MDA and HLA FEDEP

This section introduces a development lifecycle based on HLA FEDEP and MDA, and describes a five steps development framework (as shown in middle of figure 2). This new methodology has been introduced previously in (Tu 2010). It aims at adopting the strong points from both HLA FEDEP and MDA. This framework proposes five key phases for reusing existing software to achieve a rapid redevelopment of a HLA based system of systems. The task of each phase is as follows:

- Phase 1: Domain requirement definition, whose main task is to collect clear and enough requirements from customer in order to define the objective of the system, describes the environment of the system, the scenario of the system. At the same time, all these definitions and descriptions need to be reasonable, understandable for all stakeholders.
- Phase 2: Domain scenario systematization, whose main task is to refine the domain scenario and business process defined in the first phase, identifies and describes the entities involved in the scenario and business process. Then, it will define the relationships among entities and behaviours, events for each entity, etc.

- 
- Phase 3: System model specialization. In this phase, according to the technique chosen and platform selected, the system needs to be refined, for instance, to refine federation and federate structure, to allocate functions and attributes, etc. Detailed design will carry out at this time.
  - Phase 4: System Implementation, whose task is to transfer the specific system model into code, creates the executable federation and executable federate.
  - Phase 5: Test. Testing phase includes the Test Federation, Execute Federation and Prepare Outputs, and Analyze Data and Evaluate Results in HLA FEDEP. Meanwhile, it will also refer to the outputs from the previous steps, such as the original user requirement in the first step, and federation test criteria from second phase.

After this harmonization, a federate can be considered as a converter. A federate has two parts as illustrated in figure 3, one is Adapter and another is Plug-in. Adapter has Enterprise Business Behaviour Interface, which connects to the enterprise business process related to specific strategies and algorithms of different enterprises. Plug-in has Integration code, which manages the interactions between the enterprise business behaviour interface and the RTI, providing an RTI independent API to the enterprise business behaviour interface, and a simulation independent API to the RTI services.

[Figure 3 Harmonized federate structure]

The objective of these abstractions is to ensure that the enterprise business behaviour remains decoupled from RTI services. After the harmonization, all federates will have the same Integration code but different Enterprise Business Behaviour Interfaces. Meanwhile, any simulation related services required by the enterprise business behaviour interface are accessed via the integration code, rather than through direct interaction with the RTI. As a result, integration code is common components for all federates of the existing coordinators<sup>1</sup> and also the reusable components for the future coordinators. In addition, enterprise business behaviour interface will adapt to different legacy systems of different enterprises. It will be implemented based on specific strategies and algorithms of different enterprises and it will accomplish the cipher mission.

## 4.2. Model Reverse Method

This section describes a process of model reverse engineering with two different scenarios constraints. These two scenarios are presented as two arrows around the five steps life cycle as shown in the middle of figure 2. The reversal process will re-characterize the legacy system in order to capitalize on the information and functions of the existing system, and make it easy for reusing in a new HLA compliant system. This methodology will assist to HLA FEDEP / MDA alignment mentioned in previous section, to fully achieve rapid development of federation and/or federate based on the legacy IT systems.

The difference of the two scenarios constrains is the proportion of model reversal. According to the existence of HLA federation, the reversal process will stop at different steps of harmonized lifecycle mentioned in previous section. If the HLA federation has not been created yet, the model reversal process needs to start

---

<sup>1</sup> Coordinator means a member within a coordinating conjunction that indicates the HLA federation in this paper.

---

from the code of the legacy information systems to the first definition phase (domain requirement definition). If not, the reversal can stop at the second phase (Domain scenario systematization). It will only reuse the model of the existing federation to create the model for the federate related to the legacy system of new participator. All the models come from this reversal process are used to produce a federation and federate rapid development template.

A schema of model reversal structure can be seen in figure 4. This illustration is based on the approach of MoDisco (for Model Discovery) ([Jouault 2009](#)).

In MoDisco principle, a model (Mi) in the modeling world is a representation of a system in the real world and the nature of the model (Mi) is defined by its meta-model (MMi). It means that model Mi conforms to its meta-model MMi, and every step is guided by a meta-model. The very first step of a model discovery process is always to define the meta-model corresponding to the models you want to discover. Then, the second step is about creating one or many discoverers, which is illustrated in the middle of figure 4. These discoverers extract necessary information from the system in order to build a model conforming to the previously defined meta-model. The way to create these discoverers is often manual but can also be semi-automatic.

[Figure 4 a schema of model reversal structure]

In addition, in order to adapt MoDisco principle to the federated approach proposed in this paper, the “constraints” will be added onto the “discoverer” (the green box illustrated in figure 4). The “constraints” will be put before the “discoverer” (as the constraint □ shown in the figure 4, before system reversal happens) and after the “discoverer” (as the constraint □ shown in the figure 4, before the target model transformation happens) according to the following specification:

- “Constraints ①”: these constraints are used to simplify and configure the model reverse process.
  - Simplify the model reverse process: as known, the legacy system consists of lots of diverse sub-systems, which are always based on various kinds of platforms and techniques, thus it is big and only partially useful in the particular context. The reversal of the whole legacy system would be extremely huge and complicated, which departs from the objective. As a result, “constraints ①” aims at specifying the target source, which means that the bound of model reversal must be defined before start to reverse. The boundary must also be defined based on each enterprise’s confidential information. This boundary specification will be recorded as a configuration file which can be read by discoverers.
  - Configure the model reverse process: the model reverse application designed for enterprise interoperability will be applied on various enterprise systems. Thus, it must consider interoperability constraint based on the specific scenario, such as participants’ relationship, collaboration agreement, work flow, and etc. Before to execute model reverse application on different systems, the model reverse process must be configured based on the interoperability constraint.
- “Constraints ②”: these constraints are used to filter the model information obtained from model reverse tools, and guide the model transformation according to the specific requirements, such as language specific, platform specific, and so on.
  - Model information filter: this first functionality of “constraints ②” can be considered as a “filter”. Based on the current model reverse engineering technology, most of the model reverse tools can

---

obtain mass information of models. According to the different motivations, the model information might be useful or useless. The reverse method proposed in this section concerns only the system handles that provide the interfaces for data input and output. In addition, it is very complicated and dangerous to make an interoperability decision based on the complex information. Thus, it is necessary to wipe off the unnecessary information and retain only the valuable information in the considered context.

- Model transformation guide: according to the on-going research, none of the software tools can fully reverse a legacy system from code to model. Some of the tools can rewind the code to static model without the dynamic one, and some of them can only discover the data model from database. Meanwhile, the model transformation also causes the loss of information. Therefore, the obtained model information cannot be used directly for interoperability, it must be complemented. For example, in order to develop HLA components that interface with legacy IS, the behaviour models of the actions on the data also need to be discovered for implementing the mechanism for data access, the periodicity of update and the sequences of modifications accepted. Thus, the guider must complement the obtained information in order to generate the required models.

### 4.3. HLA Evolved Web Services Method

The objective of using HLA Evolved Web Services is to provide an easy-pass for the potential client to join the cooperative project. This idea is a loose couple approach. It can adapt to different potential clients with heterogeneous cooperation purposes and modalities. The detail of this method can be found in (Tu 2011)

The general idea of HLA Evolved Web Services is illustrated in figure 5. It assumes that a cooperative project has been launched. The information systems of members run properly within the federation. Assuming that during this project, other enterprises want to join this project with different expectations, such as different cooperation time periods, different cooperation domains, different expected results from federation, etc. However, to rebuild the federation is impossible. Accordingly, the solution of this approach proposes to add one more federate, who is called *WebservicesFederate* as shown in figure 5. This federate will provide various services, different access permissions, and the common API of existing federation. The candidates can use the common API and the service they prefer to generate their own federate, and then connect to the existing federation with different authorities via the Wide Area Network.

[Figure 5 HLA Evolved Web Services]

The figure 6 presents a schema of *WebservicesFederate* design proposed in this paper. *WebservicesFederate* is a special HLA federate, which is inside the Local area network (LAN) but not fully included in the HLA federation. According to this specific structure, *WebservicesFederate* is divided into two parts: one is *WebservicesBridge*, which is inside the HLA federation; another is *WebServicesServer*, which is outside the HLA federation but still inside the LAN. These two parts are connected by a socket. This design is customized for poRTIco RTI. As mentioned, this simulation is based on poRTIco RTI that doesn't support natively Web RTI functionality. In order to implement Web RTI functionality, the approach defines *WebservicesBridge* and *WebservicesServer* for *WebservicesFederate*.

- 
- **WebservicesServer**: it is used to publish web services interface to potential customers outside the federation. It takes charge of monitoring and replying to the federate via web service. When this server receives the message from the federate outside federation, it generates a User Datagram Protocol (UDP) data package and sends it to **WebServicesBridge** by the socket connection.
  - **WebservicesBridge**: it uses to synchronize the message from the federate outside the federation with other federates inside the federation. This bridge transmits messages to federate inside the federation by RTI, but exchanges messages with the **WebservicesServer** by the socket connection. When web services federation establishes, this bridge launches a thread to monitor the events happening in the web service server.
  - **Socket data package**: in order to ensure the security of the federation, common federation attributes are encapsulated into the web service interface, which is published by the web services server. So, **WebservicesBridge** encodes the attributes into the socket data package, and then this package is decoded by **WebServicesServer**. Afterwards, **WebServicesServer** generates the result that is requested by the federate outside the federation. In the opposite way, federates outside the federation can send request based on the web services it customized. While, the **WebServicesServer** receives the request, it translates the request based on the FOM, and then generates a data package which is decoded by the **WebservicesBridge**.

[Figure 6 schema of WebservicesFederate design]

#### 4.4. Short-lived ontology Method

Ontology is used to organise and handle data by semantically interconnecting them. “Short-lived ontology” is a particular non persistent ontology (Zacharewicz 2008) (Zacharewicz 2009), with a very short lifetime. To the extreme it can exist (and persist) only during a communication between interlocutors. When a message requester receives information, it will try to decode the information by using a local ontology. In case it is not possible to understand the data, it can demand to the emitter to deliver the ontology associated to this message. After the message requester obtains all the information required, the received ontology file can be deleted. However, the terms inside the ontology files can also be temporarily saved in message requester’s ontology glossary. This ontology glossary is a self-learning system with limited space (in order to save the memory and also avoid the redundancy), which means that the glossary can be self-updated automatically. In this ontology glossary, every ontology term has weighting coefficient which can measure the popularity of ontology term. In case the ontology terms got a low coefficient, it will be deleted from ontology glossary. The Figure 7 illustrates informally the communication mechanism of “Short-lived ontology”.

[Figure 7 short-lived ontology]

- a. In the case a, the “enterprise 1” sends information and the ontology to understand (decode) it at the same time. This ontology is supposed to be only valid for this information. The ontology is not persistent above the relation of the two enterprises.



- 
- b. In the case b, the “enterprise 1” sends only the information to “enterprise 2”. Once “enterprise 2” receives the information, it interprets the meaning using its local ontology if it is able to decode the information. If not, it asks for the ontology associated to the message to the sender of the message. The enterprise can conserve the new received ontology to reuse it with further data sent by the same emitter or another one also compliant to the same ontology. A “best before end” date or a countdown of validity can be associated to the ontology.

In the case a, the information can be exploited directly thanks to the ontology received at the same time. However, the information size exchanged is more important, in addition, it can be intercepted and the confidentiality can be broken. In the case b, the confidentiality is enforced but it requires more exchanges between the two partners and consequently increasing the communication duration.

According to the definition of federated approach, case b is the “on-the-fly” solution. Case b can also ensure the information confidentiality. From that postulate we introduce the concept of “short lived” ontology (this ontology definition is based on the definition in [\(Gruber 1995\)](#)), where ontology can be, in some case, suppressed after use or have a finite duration validity. This “short-lived ontology” approach will be used to handle the interoperability issue in data concern.

The authors point here that this approach works at establishing the interoperability dynamically at a semantic level. Nevertheless the data needs to be transported at the technical level. For this reason the interoperability at the physical level need to be solved also, it will be done thanks to a simulation standard described in the next section. This last level is not referring to a dynamic standard.

## 5. Case Study of the federated approach

This chapter will use a case to describe how to implement the federated approach proposed in this paper. The case describes a scenario of car purchasing and car manufacturing. This scenario includes customer ordering, car manufacturing, material purchasing/delivering, and product delivering. The actors of this case are car manufacturer, clients, semi-manufactured goods/ automobile parts suppliers, raw material suppliers, and potential participants. The actions of this case are collaboration establishment of car manufacturer and suppliers, goods order and distribution between car manufacturer and clients, material purchasing and delivering between car manufacturer and suppliers, and collaboration establishment of potential participant and existing members. The goal of this case is to use the federated approach to achieve efficient establishment of interoperability environment, rapid order dispatch, intelligent information analysis and easy pass to the federation for clients and new participants.

The federated approach proposed in chapter 4 has been implemented into a software tool. The case study mentioned in this chapter will be used to describe how to use this tool. This tool is based on poRTIco RTI and developed in Java language. It is implemented on Eclipse, and can be run in Windows NT or UNIX system with JDK 1.6.0 (or higher) environment and poRTIco environment.

---

JAX-WS (JAX-RPC) is used for implementing web services. JFreeChart is used for illustrating the simulation result.

This tool consists of two main functional modules: (1) a build time module including model reversal, model adjustment, and target model & code generation, and (2) a run time including message dispatch and management. The architecture of this tool is illustrated in figure 8.

[Figure 8 the architecture of federated approach tool]

## 5.1. Harmonization of MDA and HLA FEDEP

As mentioned in section 4.1, the harmonization of MDA and HLA FEDEP supports standardized & modularized design and development of the federated approach presented in this paper. Thus it is the guideline of the enterprise interoperability requirement analysis and the enterprise interoperability environment establishment.

- Phase 1: Domain requirement definition - The scenario of this case can be simply decomposed into (1) cars purchasing and distribution between car manufacturer and client, (2) material/ automobile parts purchasing and delivering among car manufacturer, semi-manufactured goods/ automobile parts supplier and raw material supplier, who are the initial members of this interoperation, and (3) alignment establishment between potential participant/ customers and existing members. The scenario (1) requires that clients can send the car purchasing order and trace the order process, while, the car manufacturer has to produce the car following the customer requirement and make the order process public to the clients. The scenario (2) is the common commercial processes happen when car manufacturer or semi-manufactured goods/ automobile parts supplier is in the situation of stock shortage. In this case, these processes are required to be efficient and low cost. The scenario (3) of this case is swift link and self-adjusted link for web users (potential participant and customers).
- Phase 2: Domain scenario systematization - as defined in the first phase, there are four main entities, car manufacturer, client, semi-manufactured goods/ automobile parts supplier, raw material supplier and potential participant. They are connected by various kinds of request and response functions. The use case diagram (figure 9) illustrates these entities and their relationship.
- Phase 3: System model specialization - In this case, HLA and JAVA are chosen, the scenario defined in the previous phases needs to be transferred into HLA and JAVA models. In this case, the car manufacturer and suppliers are the sponsors of the HLA federation, who start this collaboration. The clients and the potential participant will join this collaboration later after the HLA federation has been established, so they will be the HLA Evolved federate mentioned in section 4.3. All the actors join this collaboration for certain reasons such as to sell their products or purchase products, so their IT systems have their own objects for representing their interests. However, because of the federate approach, it is not proper to define common attribute for the objects of everyone's "interests". Thus, in the HLA context, a JAVA object of HLA message is

---

defined for “interests” with attributes of sender, receiver and message context. All the members are connected by various kinds of request and response functions, so different HLA interaction classes are defined for these function, such as, the “Material Delivering” HLA interaction class. Inside these classes, there are the parameters of HLA specific timestamp and message object defined before.

- Phase 4: System Implementation - the objects and interaction classes defined in phase 3 will be generated into HLA FOM file, so that it will be part of the plug-in (“integration code”) mentioned in section 4.1.
- Phase 5: Test - the test case will be prepared and be used in each phase’s evaluation and final validation.

[Figure 9 Use case diagram]

## 5.2. Build Time I

The Build Time I is responsible for interoperability environment establishment. It means that Build Time I must bring all the participants’ existing IT systems together for Enterprise Interoperability. Thus, the main task of Build Time I is to discover models from legacy systems, and perform the interoperability modelling on these models.

### 5.2.1. Analyze UML

Firstly, we use MoDisco Tool (Bézivin 2006) to discover UML models from legacy system. As mentioned in 4.2, not all the reversed models are useful, so this tool provides a “filter” module called “UML Reader and Analyst” (as shown in figure 10) to load the UML information. This module will load the UML class information and illustrate it into the hierarchy, “root -> package -> class -> attribute”, as the column name of the table shown in figure 10. After loading the reversed models, this application will provide a well arranged view of class relevant information as the table shown in figure 10. Afterwards, user can delete the unnecessary UML class information by selecting the row or the cell where it locates in. After clicking button “save” to confirm that the rest information in the table is the expected information, the rest information will be saved into an xml file. This xml file is the input of the module “model alignment” that will be explained in next section, so it only retains the classes with their attributes. It can also be considered as the first version of the HLA SOM of the correlative enterprise federate.

[Figure 10 UML Reader and Analyst Application]

### 5.2.2. Model Evolution & Model Alignment

As mentioned in section 4.2, the reversed models must be well organized to reduce the complexity of model analysis. Thus, this tool provides a function module called “model evolution and model

---

alignment”. Model alignment is responsible for aligning the similar models of different enterprises into a new model. Model evolution is responsible for categorizing the enterprises into different groups for model alignment based on their business relationship.

The figure 11 shows the user interface of the model alignment module. Firstly, model evolution module will load the xml files (the output of module “UML Reader and Analyst”) of the enterprises from one group. And then, enterprises’ models will be illustrated on a table as a symmetric relation matrix, which means the column and row of this table represent the same information. Afterwards, if the user selects one cell of the table, then the class diagrams of the correlative row and column will show out in the class diagram illustration area. By comparing the class diagrams, user can define the similarity for the selected classes. As the symmetric relation matrix has a feature called relation transmission (Wang 2000), which defines that if Matrix M has  $M_{ij} = 1$  (it means that i and j has R relation) and  $M_{jk} = 1$ , then  $M_{ik} = 1$ . Thus, the cells of this table are bound to a trigger event that will require program to check all the cells on the corresponding row and column in order to detect the possibility of similarity transmission. Because the similar relation is not always transmittable, the program only highlights the possible cells to suggest user to determine.

When the user finish the model alignment, he can click the save button to confirm. And then, under the class diagram illustration area, a new operation panel (as shown in the figure 11, within the blue box) shows up for aligning the model structure. It is rename functionality by listing out the groups of classes and their attributes. The user can delete or rename the attributes, and input the new name for the classes in the text field at the bottom.

[Figure 11 user interface of the model alignment module]

After “model evolution and model alignment”, the set of similar models can be used to initiate the HLA FOM file according to the correlative RTI format. For example, this simulation chooses the portico RTI, then, the HLA FOM file will be organized in the format shown in the figure 12.

[Figure 12 portico RTI based HLA FOM file]

### 5.3. Run Time

Run time is responsible to simulate the enterprise interoperability in a HLA environment. The figure 13 shows an example of the user interfaces. The user interface of car manufacturer federate within the red box on the top left is for automobile parts purchasing. For example, the car manufacturer wants to order automobile parts from suppliers. He can input the information, such as product name, product category, and demand. Afterwards, this request will be sent to the RTI who will dispatch it to automobile parts suppliers’ federates. The automobile parts suppliers’ federates will query the database based the request information automatically (as the user interface within the green box on the bottom of figure 13). If the automobile parts suppliers have enough inventories of required goods, they will answer this requirement. Then, the federate of car manufacturer will analyse all the

---

feedbacks, decode the messages, and then generate an analysis report. The analysis report is shown in the user interface within the blue box on the top right of figure 13. The report includes the names of the suppliers, their location, the distance from car manufacturer to them, the inventory number, the price and etc.

[Figure 13 Federate user interface]

The application also provides a clearer view of the summary of total cost based on the data provided by this report (as shown in figure 14). The application can calculate the sum of the goods' price and the freight charge, so that the manufacturer can visually compare the suppliers. Finally, the manufacturer can decide to purchase the materials with which automobile parts suppliers. For example, the figure 14 shows that the cost of purchasing from supplier in Bordeaux is most reasonable.

[Figure 14 Analysis Result]

## **5.4. Build Time II**

Build Time II is responsible for interoperability establishment between potential participants from web and the existing interoperability environment.

### **5.4.1. Generate federate of potential participant**

The illustration of the user interface of alignment establishment between potential participant and existing members is shown in figure 15. For example, if one supplier wants to be the member of the collaboration project, he can find the Web Services published by WebservicesFederate (as the user interface within the red box on the right side of the figure 15 shown). Afterwards, he can choose the collaboration items and fulfil his own information, and then submit the application form to the existing federation. The administrator of the HLA federation will verify the applicant list (as the user interface within the green box on the left side of the figure 15 shown), and decide which application can be accepted. After passing the evaluation, the applicant can download the Web Services package to generate his own federate interface.

[Figure 15 Alignment establishment between potential participant and existing members]

In addition, the new participant will perform the ontology alignment with the existing participants in the HLA federation to create his local short-lived ontology glossary, which will be used for automatically analyze the information from existing HLA federation. The table 1 shows an example of the result of the ontology alignment that uses the multi-strategies ontology alignment approach (Song 2012). This approach will assign similarity value to each pair of ontology elements. Only the pairs with the similarity value bigger than the threshold value will be kept and aligned.

---

[Table 1 Ontology alignment between new participant and existing participants]

### 5.4.2. Interaction of federate of potential participant

The figure 16 shows an example of car purchasing and car manufacturing which happen between a customer from web and car manufacturer federate with HLA federation.

Firstly, the customer sends an order to the car manufacturer federate (CMF). And then the car manufacturer federate calculates the amount of raw materials based on the bill of material. After that it dispatches orders to the different suppliers, such as wheel supplier (WS) and engine supplier (ES), to get the parts. When suppliers finish the production, they deliver the products to the CMF who assembles them and then deliver to the customer. In this simulation, the status of the order is the concerned issue. The customer cares about when he can receive his cars and the status of this manufacturing process. As a result, the customer does not have the complex interaction with the existing federate within the HLA federation. Thus, he does not need to follow the complex procedure as the web applicants mentioned earlier. The only thing needs to know is what kinds of cars the car manufacturer can provide. The only thing needs to follow is the HLA service Interface. By implementing this interface, the customer can send the purchasing order, can know when he can receive the cars, and can trace the manufacture process.

[Figure 16 Example of car purchasing and car manufacturing]

As the user interface within the red box on the right side of the figure 16 shown, the customer can input order quantity. After the confirmation of this order, the client side shows the total number of manufacturing days. During the car manufacturing, the customer can request the detail of the manufacturing process, and the federation immediately sends back the result, which has been generated as a bar chart to vividly illustrate the status of the manufacturing process. In the CMF user interface (as the user interface within the green box on the left side of the figure 16 shown), the remaining days of car manufacture is presented. In order to assist the car assembling process, the information of the suppliers is presented. By clicking the 'refresh' button, CMF can receive the latest information from each supplier. The Gantt chart shown in figure 17 is a real time report of the car manufacturing schedule. As shown, the car assembling process is waiting for the wheel and engine, any delay caused by suppliers will postpone the schedule of car assembling process. Thus, the car manufacturer has to monitor the suppliers' manufacturing process, in order to be able to cope with the incidents by adjusting the schedule or proposing a new solution. This Gantt chart allows the car manufacturer to monitor the suppliers in real time. For example, the figure 17 shows that the wheel supplier has completed 20% process, and engine supplier has completed 40%, both of them are progressing under the schedule.

[Figure 17 Car manufacturing schedule]

---

## 6. Conclusion and Future work

This paper has proposed a federated approach to develop enterprise interoperability dynamics. This approach has presented the development framework for HLA based platform, which can accelerate the establishment of virtual enterprise collaboration, and also provide the “easy pass” service for adapting to different potential clients with heterogeneous cooperation purposes and modalities. This framework includes model reverse engineering and HLA Evolved approach. This framework provides a five step process to generate models for simulation starting from conceptual enterprise models to be converted to MDA models and code to accelerate the rebuilding of legacy information systems for implementation of information system exchange facilities. This method works at bridging the gap from concepts to implementation in the field of enterprise Information Systems development. The long-time experience in management, interoperability and synchronization of data of distributed simulation is reused by applying HLA standard for information exchange. The HLA Evolved approach has been carried out with the open source RTI, poRTIco extended by a new component straddling between HLA federation LAN and WAN to fulfil HLA 1516-2010 requirements. After passing some gaps between HLA Evolved approach requirements and the 1.3 API provided by poRTIco, the simulation runs properly for different test cases elaborated within laboratory data.

This approach is still an on-going research. Up to now, it focused on the data and service levels, thus cannot fully cover all the multidimensional concerns in Enterprise Interoperability Framework of INTEROP NoE. Meanwhile, because the solution of this approach is based on HLA technology, the establishment of dynamic interoperability still has a common standard to follow even if it is only in the technical level. Even so, this research work can be considered as an answer to new challenges engendered by future internet requirements at the semantic level, and to create in particular enterprises more dynamically interoperable. It is also one of the concrete implementations of what is specified in the new release of the HLA 1516-2010 standards. To finish, as an initial step validation, the simulation has passed the test to run on academic data. Nevertheless it still needs to be improved to support large quantity of industrial data. This opportunity will be given by a European FP7 Future Internet project the authors are involved in.

## Reference

- ABILITIES project, 2008. *Application Bus for Interoperability In enlarged Europe SMEs*, Available from <http://www.dbis.cs.uni-frankfurt.de/index.php> [accessed 07 July 2011].
- Bourey, J.P., Grangel Seguer, R., Doumeings, G., and Berre, A.J., 2007. *Report on Model Driven Interoperability*, Deliverable DTG 2.3, INTEROP NoE, April, pp. 91. Available from: <http://www.interop-vlab.eu/> [accessed 15 June 2009]
- Bézivin J., Brunelière H., Barbero M., 2006. *The Model Discovery (MoDisco) Component: A Proposal for a New Eclipse/GMT Component, Version 1*. INRIA (ATLAS Group). Available from <http://www.eclipse.org/MoDisco/about.php>.

- 
- Charalabidis Y., Gionis G., Moritz Hermann K. and Martinez C., 2008. *Enterprise Interoperability Research Roadmap, Draft Version 5.0*. Available from [ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/enet/ei-roadmap-5-0-draft\\_en.pdf](ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/enet/ei-roadmap-5-0-draft_en.pdf) [accessed 20 December 2010]
- Chen D., Shorter D., 2008. Framework for Manufacturing Process Interoperability. *Standards for Interoperability - How, Workshop in Conjunction to I-ESA*, Berlin : Germany.
- Chen D., 2009. Framework for enterprise interoperability. *Congrès International de Génie Industriel (CIGI2009)*, Bagnères de Bigorre: France.
- COIN Project, 2011. *Enterprise Collaboration & Interoperability*. Available from <http://www.coin-ip.eu/> [accessed 07 July 2011].
- Doumeings G., Ducq Y., 2001, Enterprise Modelling techniques to improve efficiency of enterprises, *International Journal of Production Planning and Control - Taylor & Francis*, Vol. 12, number 2, pp 146-163
- EIF, 2011. *European Interoperability Framework (EIF) Towards Interoperability for European Public Services*. Available from [http://ec.europa.eu/isa/documents/eif\\_brochure\\_2011.pdf](http://ec.europa.eu/isa/documents/eif_brochure_2011.pdf) [accessed 07 August 2013].
- Elvesæter B., Hahn A., Berre A., Neple T., 2007. Towards an Interoperability Framework for Model-Driven Development of Software Systems. *Interoperability of Enterprise Software and Applications*, 409-420.
- ENSEMBLE, 2011. *Deliverable 2.1 EISB State of Play Report version 1.00*. Available from <http://www.fines-cluster.eu/fines/wp/d21/> [accessed 07 July 2011].
- FUSION project, 2008. *Business process FUSION based on Semantically-enabled Service-Oriented Business Applications*, Available from <http://www.fusionweb.org/> [accessed 07 August 2013].
- GENESIS, 2006. *Enterprise application interoperability via Internet-Integration for SMEs, Governmental Organisations and Intermediaries in the new European Union Deliverable D1.1*. Available from [http://interop-vlab.eu/ei\\_public\\_deliverables/genesis-deliverables](http://interop-vlab.eu/ei_public_deliverables/genesis-deliverables) [accessed 07 August 2013].
- Gorka B., Larrucea X., Elvesæter B., Neple T., Beardsmore A., Friess M., 2007, *A Platform Independent Model for Service Oriented Architectures*, Enterprise Interoperability, Ed, London, S., pp. 23-32. ISBN 978-1-84628-713-8.
- Gonçalves R., Agostinho C., Garção A., 2012. A reference model for sustainable interoperability in networked enterprises: towards the foundation of EI science base. *Computer Integrated Manufacturing*. 25(10), 855-873.
- Gruber T., 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6), 907 - 928.
- IDABC, 2008. *European Interoperability Framework draft version 2.0*. Available from <http://ec.europa.eu/idabc/servlets/Docb0db.pdf?id=31597> [accessed 07 July 2011].
- IEEE std 610, 1990. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, Institute of Electrical and Electronic Engineers.
- IEEE std 1516.2-2000, 2000. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification*, New York: Institute of Electrical and Electronic Engineers.
- IEEE std 1516.3-2003, 2003. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federation Development and Execution Process (FEDEP)*, The Institute of Electrical and Electronic Engineer.
- IEEE std 1516<sup>TM</sup>-2010, 2010. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -- Framework and Rules*, New York: Institute of Electrical and Electronic Engineers
- IEEE std 1730<sup>TM</sup>-2010, 2011. *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*, The Institute of Electrical and Electronic Engineer.



- 
- Jouault F., Jean B., Mikael B., 2009. Towards an advanced model-driven engineering toolbox. *Innovations in Systems and Software Engineering*, volume 5, pp 5-12.
- NisB project, 2010. *The Network is the Business*, Available from <http://www.nisb-project.eu/index.php> [accessed 07 July 2011].
- OMG, 2003. *MDA Guide Version 1.0.1*. Object Management Group, Document number: OMG / 20030601. Available from: [www.omg.org/docs/-omg/03-06-01.pdf](http://www.omg.org/docs/-omg/03-06-01.pdf) [accessed 15 June 2009].
- OMG, 2010. *Architecture Driven Modernization (ADM): Knowledge Discovery Meta-Model (KDM) v1.2*, OMG, Document number: formal/2010-06-03. Available from: <http://www.omg.org/spec/KDM/-1.2> [accessed 15 July 2010].
- OMG, 2011. *Architecture-driven Modernization: Abstract Syntax Tree Metamodel (ASTM) v1.0*, OMG, Document number: formal/2011-01-05. Available from <http://www.omg.org/spec/ASTM> [accessed 25 April 2011].
- poRTIco, 2009, *Developer Documentation*. Available from: [http://porticoproject.org/index.php?title=-Developer\\_Documentation](http://porticoproject.org/index.php?title=-Developer_Documentation) [accessed 15 July 2010]
- Richardson L., Ruby S., 2007, *RESTful web services*, O'Reilly Media, Inc. p. 299
- Song F., Zacharewicz G., Chen D., 2012. Multi-strategies Ontology Alignment Aggregated by AHP. *16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, IOS Press, San Sebastian, 1583-1592.
- Touzi J., 2007, *Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises*, PHD thesis, Institut National Polytechnique de Toulouse.
- Tu Z., Zacharewicz G., Chen D., 2010. Unified Reversible Life Cycle for Future Interoperable Enterprise Distributed Information Systems. *IESA 2010-Interoperability for Enterprise Software & Applications 2010*, 57-66, April 14-15, Coventry (UK).
- Tu Z., Zacharewicz G., Chen D., 2011. DEVELOPING A WEB-ENABLE HLA FEDERATE BASED ON PORTICO RTI. *Proceedings of the 2011 Winter Simulation Conference*, Orlando USA.
- Wang Y., 2000. *Discrete Mathematics Introduction*. Harbin Institute of Technology Press.
- Zacharewicz G., Chen D., Vallespir B., 2008. HLA Supported Federation Oriented Enterprise Interoperability, Application to Aerospace Enterprises. *Proceedings of 2008 International Simulation Multiconference EuroSISO*, 08E-SIW-074, June 19-19, Edinburgh (Scotland).
- Zacharewicz, G., Chen, D., Vallespir, B., 2009. Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability, *Proceedings IEEE of International Conference I-ESA China 2009 Interoperability for Enterprise Software and Applications*, pp. 329-335, April 22-23, Beijing (China)