HARMONIZED AND REVERSIBLE DEVELOPMENT FRAMEWORK FOR HLA BASED INTEROPERABLE APPLICATION

Z. Tu(a), G. Zacharewicz(b) D. Chen(c) and Z. Zhang(d)

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

[(a)]zhiying.tu@ims-bordeaux.fr, [(b)] gregory.zacharewicz@ims-bordeaux.fr, [(c)] david.chen@ims-bordeaux.fr, [(d)] Zhenchuan.Zhang.Simon@gmail.com

ABSTRACT
This paper aims at improving the re-implementation of existing information systems when they are called to be involved in a system of systems, i.e. a federation of enterprise information systems that interoperate. The idea is reusing the local experiences coming from the development of the original information system with the process of Model Discovery and Ontological approach. We draw the strong points of MDA and HLA, former in transforming concepts and models from the conceptual level to the implementation and latter in implementation of distributed systems, then we propose a MDA/HLA framework to implement distributed enterprise models from the conceptual level of federated enterprise interoperability approach. In addition, we propose a model reversal methodology to help re-implement the legacy information system, in order to achieve the interoperability with other systems. We also draw the strong points from web services in order to improve the performance of HLA in distribute simulation.

Keywords: Interoperability, HLA, Model reversal, HLA evolved

1. INTRODUCTION
In the globalised economic context, the competitiveness of an enterprise depends not only on its internal productivity and performance, but also on its ability to collaborate with others. This necessity led to the development of a new concept called interoperability that allows improving collaborations between enterprises. No doubt, in such context where more and more networked enterprises are developed; enterprise interoperability is seen as a more suitable solution to total enterprise integration.

Since the beginning of 2000, several European research projects have been launched to develop enterprise interoperability (IDEAS, ATHENA, INTEROP). Three main research themes or domains that address interoperability issues were identified, namely: (1) Enterprise modeling (EM) dealing with the representation of the internetworked organization to establish interoperability requirements; (2) Architecture & Platform (A&P) defining the implementation solution to achieve interoperability; (3) Ontologies (ON) addressing the semantics necessary to assure interoperability.

This paper will contribute on rapid and intelligent development of Distributed Enterprise Information Systems by proposing a harmonized and reversible development framework for HLA based interoperable application. In fact, A multiple agent/HLA enterprise interoperability methodology and an unified reversible life cycle for interoperable enterprise distributed information systems have been proposed in (Zacharewicz 2008) & (Tu 2010). They mention that several enterprises, who participate in a cooperative project, need to exchange various information any time through a platform based on HLA. However, most of the enterprises have their own legacy information systems and the question is how to use HLA to connect them conveniently, rapidly, flexibly and perfectly solve the interoperability issues. This paper will use the harmonization of MDA (Model Driven Architecture) and HLA to form a rapid and flexible development life cycle. In order to discover the enterprises' knowledge quickly, Model Reverse Engineering will be used to reverse the legacy information system of enterprises backwards to models, and concurrently to solve the interoperability problems without rebuilding the information systems or integrate them (Jouault 2009). HLA and Web Services will be mixed in order to improve HLA's performance, which enables a closer integration of simulation based systems with traditional operational systems in a systems-of-systems approach. Web Services will help HLA to achieve Distributed Simulation Interoperability with external Systems in world wide web (Möller 2005).

The paper is structured as follows. We start out with a description of background, explain the problems we want to solve (section 2). Then we give a survey of related work in the area of HLA, MDI, Web services and Model Reverse Engineering (section 3). Then, we describe our framework in general and related scenario (section 4). After that, we explain our methodology separately in three parts, harmonization of MDA and HLA FEDEP, model reversal and HLA Evolved Web services (section 5).

2. PROBLEM STATEMENT

The enterprise interoperability framework mentions that there are three barriers for enterprise interoperability, conceptual barrier, technological barrier, organizational barrier (Ullberg 2007). And those barriers are the problems we want to solve. We propose a synthetic development life cycle to facilitate the development of the dynamic, secure and synchronized interoperable HLA based platform for enterprises with heterogeneous legacy IT system.

The three crosses in figure 1 represent three barriers in enterprise interoperability framework. For example, *conceptual barrier*, different IT systems use different terms to describe "car"; *technological barrier*, different IT systems use different data format for message transmission; *organizational barrier*, diverse organization structures exist in different enterprises. In addition, there is another problem. When an enterprise intents to participate in a existing cooperative project and connect to other heterogeneous IS, how can we establish this connection efficiently, correctly, and with low cost? In this paper, we are going to answer this question.

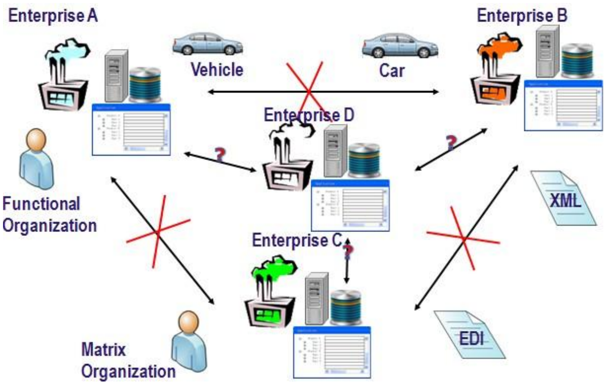

Figure 1 Problems to solve

3. TECHNICAL RECALL

3.1. Model-Driven Interoperability framework

The approach "Model Driven Interoperability" (MDI) consists in considering interoperability problems from enterprise models level instead of only at the coding step.

These works were realized in the Task Group 2 (TG2) of INTEROP-NoE state at defining an approach inspired from OMG MDA (Bourey 2007). The goal is to tackle the interoperability problem at each abstraction level defined in MDA and to use models transformations techniques to link vertically the different levels of abstraction or horizontally to ensure each models of the level interoperability. The main goal of MDI, based on model transformation, is to allow a complete follow from expressing requirements to coding of a solution and also 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 (Chen 1997) (Doumeingts 2001) Models to UML between CIM and PIM levels (Bourey 2007). These works are additional works realized in the context of ATHENA to define UML profiles to take into account Service Oriented Architectures at PIM level (Gorka 2007). These results have consolidated by results presented by (TOUZI 2007) that have proposed an interoperability transformations method from BPMN to UML in the context of services oriented architecture.

Nevertheless, the soundness of the methodology has been demonstrated, but no full industrial scale validation have been yet realized. A project ISTA3 has especially for goal to demonstrate these concepts in an industrial real world significant application. The different methodological propositions will be tested and refined by focusing on model interoperability and their interoperability. It will consist in particular to improve the flexibility of the MDI transformation process in the way of obtaining dynamic interoperability in the context of federated approach..

3.2. 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 Defense Modeling and Simulation Office (DMSO) of the US Department Of Defense (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.

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 spiral development of the simulation system through phases of requirements, conceptual modeling, design, software development, integration, and execution (IEEE 2003).

3.3. MDA & Model Reverse Engineering

MDA is well-known for promoting the use of models and their transformations to consider and implement different systems. After MDA became an important change in software development practice, there is another research soon launched in OMG, later called Architecture Driven Modernization (ADM) (OMG 2008).

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). When the PSM is obtained, then we still need to generate code (OMG 2003). In the opposite aspect of view, ADM is trying to track the models by backwards to discover models from legacy systems. However, sometimes one would like to discover more specific models from a legacy system. This is why the ADM group has defined several meta-models to this purpose, the best known being the knowledge discovery meta-model (KDM) and abstract syntax tree meta-model (ASTM) (OMG 2008).

KDM is a meta-model for representing existing software system and the associations, relationships among the function models in the system, and also describe the operation environments. It can insure the interoperability among the existing systems, make the data exchange among different vendors' tools easier.

3.4. Web Services

Web Services is a well known technique in distribute interoperability simulation domain. It has achieve a great success in business domain, which stems from the good characteristics of the technology itself, and widely recognized by enterprises and business organizations and effective support for the open source community (Richardson 2007).

The data exchange of Web services is based on open standards such as HTTP and XML, is not associated with any particular vendor, operating system and programming language, which makes Web services platform with a good vendor neutrality, coarse-grained business functions can also be packaged for the Web service, and it can 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, and World Wide Web and other standardization organizations active participation in the Web services technology provides great maturity and popularity of the organization advantage.

4. OVERVIEW OF FRAMEWORK

4.1. Framework Description

Harmonized and Reversible development framework for HLA based interoperable application, is the general description of this framework. In this title, there are four keywords, harmonized, reversible, HLA, and interoperable.

Harmonized means this framework is a synthetic framework, which consists several techniques. As figure 1 shows, in this framework, we create a new five steps development life cycle which aligns MDA and HLA FEDEP. MDA is easy to use and easy to understand, and tightly bounding with Unified Modeling Language , Meta-Object Facility (MOF), and Common Warehouse Meta-model (CWM). It is the best choice to overcome the interoperability barriers, which mentioned in MDI framework (Elvesæter 2007). HLA FEDEP is the standard of development and execution of HLA federation, and it is similar with the waterfall development but with look-back test phrase. The motivation of alignment of MDA & FEDEP

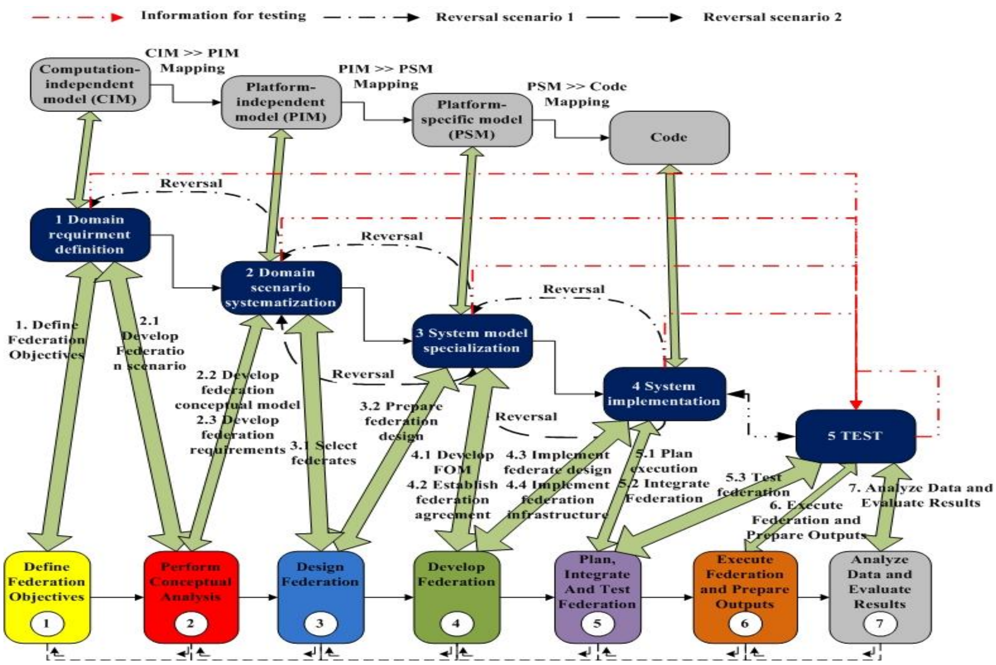

Figure 2 Harmonized and reversible development framework

is they have lots of similar steps and tasks and easy to reconstruct and align. In addition, this framework also use web services to improve the performance of HLA, which will explain in section 5.3.

Reversible means that this framework uses model reverse engineering technique to discover the model from the legacy system. To involve model reverse engineering technique aims to avoid to rebuild the legacy system for different cooperation. Our objective is to accelerate the development and reduce the cost. As figure 2 illustrates, there are two arrows, which have opposite direction to five step development life cycle. These two arrows represent two different scenarios of model reversal in this framework. We will explain this reversal in 5.2.

HLA means that this framework dedicates to the development of HLA based application. In our project, we choose an open source RTI, PORTICO. The reason of choosing that, is not only because it is free, but also we want to initiate an open framework to receive more comments and contributions from people who are interested in this idea.

Interoperable means that this framework provides a solution for achieving enterprise interoperability. As mentioned in section 2, we are going to overcome the enterprise interoperability barriers and help to realize the short-lived ontology approach, which mentioned by (Zacharewicz 2009).

4.2. Scenario

A schema of the related scenario is shown in figure 3. We assume that before enterprises start to launch a cooperative project, they all have their own system. Thus, our goal is to achieve the interoperability among those legacy systems. The steps of our approach as follows.

Step 1: model reverse engineering will be used to discover the models from the legacy system based on enterprises' requirement and our interest. And then, rewind these models into MDA models based on the alignment of MDA and HLA FEDEP, and solve interoperability problem of each level of MDA models according to the principle of MDI framework.

Step 2: a test of the final models we got by model reverse engineering is carried out. After that, we are going to transform correct models from CIM to code, and generate a Federate Interface, which can plug into the HLA platform and transmit the information with other companies' information systems via RTI.

Step 3: if there are other enterprises want to join this cooperative project, they also need to follow the step 1 and step 2, to rewind their legacy system into MDA models, and remove the incompatible parts, then generate the Federate Interface, finally, synchronize with other systems.

5. SPECIFICATION OF FRAMEWORK

5.1. Harmonization of MDA and HLA FEDEP

As known, HLA has lot of advantages, such as generalized development process: Federation Development and Execution Process (FEDEP), synchronization standard: Runtime Infrastructure Specification, Data Standards and etc, however, in order to keep these advantages, HLA needs to benefit from the commercial developments. As MDA is popular, widely recognized and it is an align able development life cycle with HLA FEDEP, and MDA can facilitate the construction of simulators and provide the standardized meta models to this integration (Andreas 2002) (Shawn 2003) (Trbovich 2005), MDA will be the best choice for HLA to benefit from. Besides that, from interoperability aspect of view, because most of the enterprises prefer to build their information system by using MDA, MDA would be the best choice for overcoming the interoperability barriers (Ullberg 2007). Model-Driven Interoperability (MDI) Framework provides a foundation, consisting of a set of reference models, for how to apply Model Driven Development (MDD) in software engineering disciplines in order to support the business interoperability needs of an enterprise (Elvesæter 2007).

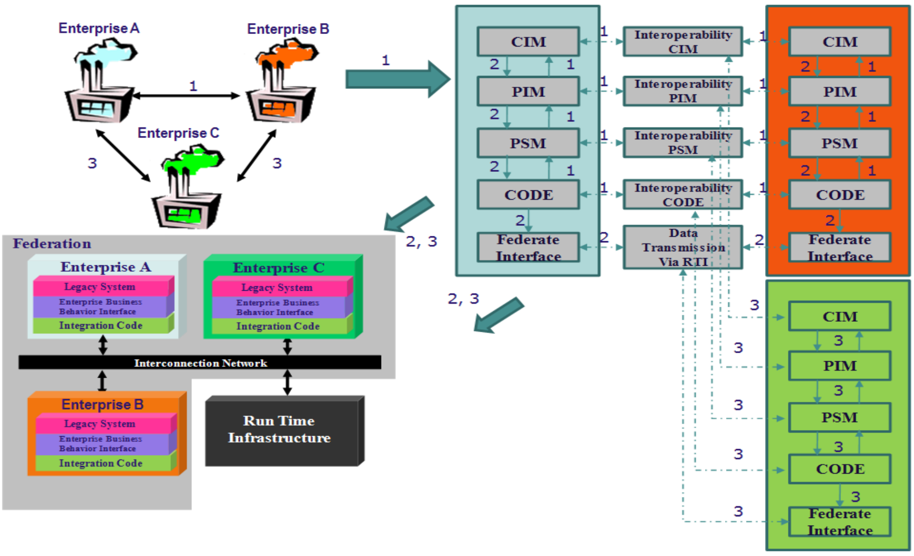

Figure 3 Scenario description

5.1.1. Specification of the harmonization

In this section, we propose a new development lifecycle to reconstruct HLA FEDEP and MDA, and generate a new five steps development framework (as shown in figure 4). This new methodology aims to adopt the strong points from both HLA FEDEP and MDA while overcoming their weak points, then, to achieve proper component reuse and rapid development.

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, to describe the environment of the system, the scenario of the system. At the same time, all these definition and description need to be reasonable, understandable for all the stakeholder. CIM of MDA has more similar task with Define Federation Objectives, Develop Federation scenario together in HLA FEDEP. As the result, we align them in this phase, to convert the user requirement, which is more textual based, into more visual model, such as UML use case to derive the federation requirement, etc.

Phase 2: *Domain scenario systematization*, whose main task is to refine the domain scenario and business process defined in the first phase, to identify and describe the entities involved in the scenario and business process. And then, to define the relationships among entities and behaviors, events for each entity, etc. This phase integrates PIM in MDA, which describes the operation of system but doesn't address the detail platform information yet, as well as steps of Perform Conceptual Analysis, Develop Federation Requirements and Select Federates in HLA FEDEP, which also define and select general participators of the federation, then describe their relationship, behaviors and event in general.

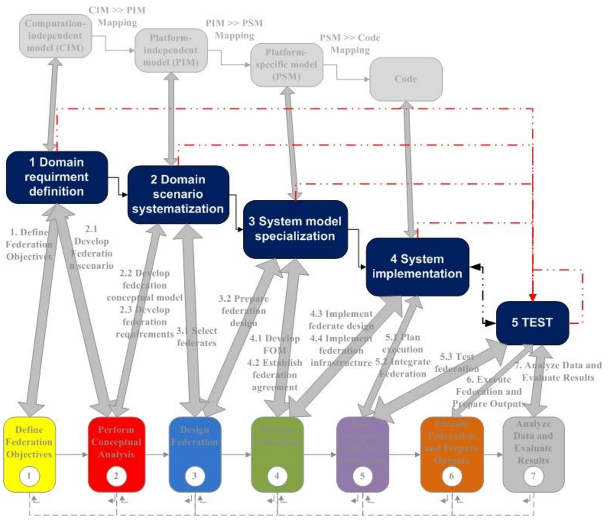

Figure 4 Harmonization of MDA and HLA FEDEP

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. This phase integrates the following parts in MDA and FEDEP. PSM in MDA, which is in the form of software and hardware manuals or even in an architect's head, will be based on detailed platform models, for example, models expressed in UML and OCL, or UML, and stored in a MOF compliant repository. The Prepare federation design, Prepare plan, Develop FOM, and Establish federation agreement in FEDEP will produce federate responsibilities, federation architecture, supporting tools, integration plan, VV&A plan, FOM, FED/FDD and time management, date management, distribution agreements, etc.

Phase 4: *System Implementation*, whose task is to transfer the specific system model into code, to create the executable federation and runable federate. At this level, MDA has various transformation techniques from model to code. In the FEDEP, Implement Federate designs will provide modified and/or new federates and supporting database. Implement Federation Infrastructure will provide implemented federation infrastructure and modified RTI initialization data. Plan Execution and Integrate Federation will provide execution environment description and integrated federation.

Phase 5: *Test*. Throughout the previous steps of the MDA and HLA FEDEP alignment process, testing is essential to ensure fidelity of the models. 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.

5.1.2. Harmonized federate structure

A schema of the expected result can be seen in figure 5. After this harmonization, we can consider a federate as a *converter*. A federate has two parts, one is *Adapter* and another is *Plug-in*. *Adapter* includes *Enterprise Business Behavior Interface*, which connects to the enterprise business process related to specific strategies and algorithms of different enterprises. *Plug-in* includes *Integration code*, which manages the interactions between the enterprise business behavior interface and the RTI, providing an RTI independent API to the enterprise business behavior interface, and a simulation independent API to the RTI services.

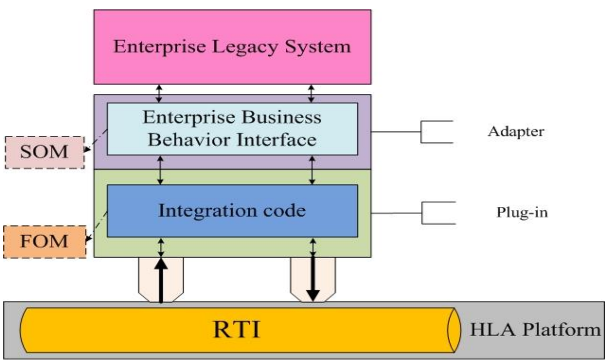

Figure 5 Harmonized federate structure

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

5.2. Model Reversal

5.2.1. Specification of Model Reversal

This section describes a brand-new process of model reverse engineering with different scenarios constraints (see figure 6). The reverse 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.

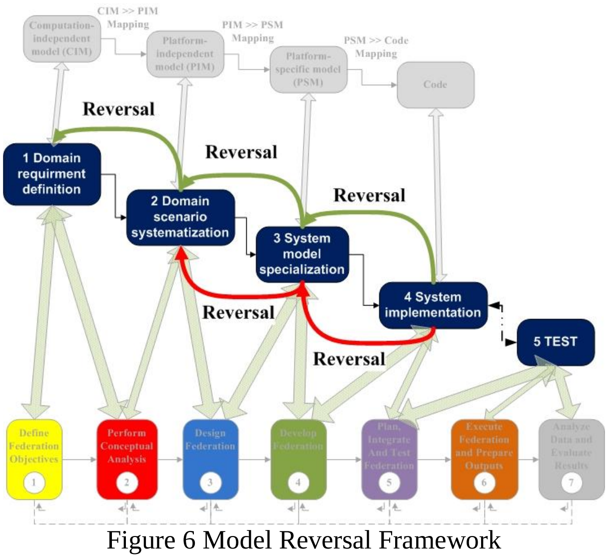

Figure 6 Model Reversal Framework

There are two different arrows, which represent two different scenarios of the model reverse process, illustrated in figure 6.

1. First, when an enterprise intents to start exchanging information in a new cooperative project with other enterprises. In that case, the HLA federation has not been created yet, so we propose to reverse the code of the legacy information systems to the first definition phase (domain requirement definition). Then from top to down, we generate the model for each phase, finally we produce a federation and federate rapid development template.
2. Second, if an enterprise intents to participate in a existing cooperative project and exchange data with other heterogeneous IS. Thus, we assume an HLA federation has already been created. Here, according to the HLA FEDEP, federate starts to be considered from the second step (Perform Conceptual Analysis) as the reversal scenario 2 shows in fig. 3.2. Therefore it is not necessary to reverse to the first phase, the reversal can stop at the second phase (Domain scenario systematization). One will only reuse the model of the existing federation to create the model for the federate related to the legacy system of the new participator. Finally, the model of the existing federation and the new federate model are used to generate the code template for the new federate for rapid development.

5.2.2. Expected result of Model Reversal

A schema of the expected result can be seen in figure 7. This illustration is based on the approach of MoDisco (for Model Discovery).

In MoDisco principle (Jouault 2009), 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 7. 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.

In addition, in order to adapt MoDisco principle to our approach, we add *constrains* before the *discoverers*. As we know, the legacy system consists lots of subsystems, which are always based on various kind of platforms and techniques, thus it is big and complex. If we are going to reverse the whole legacy system, the project will be extremely hug and complicated, and it departs from our motivation. As the result, constrains aims to specify the target source, which means that we must first define the bound of model reversal before start to reverse. And the boundary must be defined based on our interest and each enterprise's confidential information. This boundary specification will be recorded as a configuration file which can be read by *discoverers*.

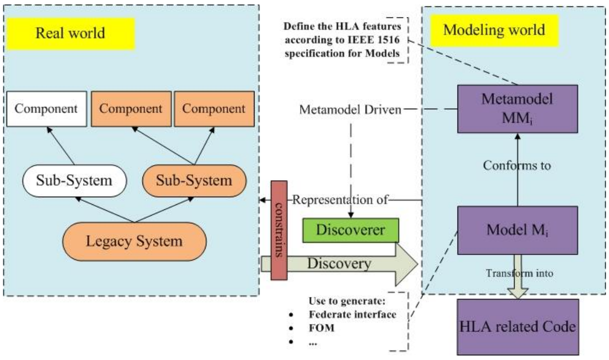

Figure 7 The expected result of model reversal

According to the ongoing research, none of the software tools can fully reverse the 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. In our approach, we choose MoDisco, which is an Eclipse GMT component for model-driven reverse engineering. The common infrastructure of MoDisco is inspired by the KDM, which provides a comprehensive high-level view of application behavior, structure, and data and corresponds to a coarse-grained representation of a given legacy portfolio. MoDisco is a powerful tools even though it needs to be further improved. It provides different kinds of DSLs (associated to a meta-model) to describe the facet you want to discover from a given legacy. Furthermore MoDisco provides a toolbox that facilitates the building of model discoverers from the definition of the target meta-models. MoDisco can generate static model from the code of program, and can also generate dynamic model from the execution of program. However, the output of MoDisco is not adaptive to our project. Now, the output is a XML based file and with a tree structure illustration on eclipse, it is not easy for the people without strong IT background to understand it. So, our objective is based on MoDisco, visualize its output and add constrains in order to adapt to our project.

5.3. HLA Evolved Web Services

In the past few years, HLA has succeeded in many aspects, especially in the areas of interoperability and reuse, however, in pace with the rapid technical change and the future of the standard, HLA has to face to many new challenges.

5.3.1. Specification of HLA Evolved Web Services

As known, HLA has been developed within the defense simulation community, and more focus on the simulation in Local Area Network or in Virtual Private Network, thus disadvantages expose: lack of flexibility, lack of integration with business and etc. Meanwhile, Web Services has been developed within the commercial enterprise community, and has achieved a great success in business domain, which stems from the good characteristics of the technology itself, and widely recognized by enterprises and business organizations and effective support of the open source community. As the result, Web Services is the perfect option to help HLA overcome those disadvantages.

The general idea of HLA evolved Web services is illustrated in figure 8. We assume that a cooperative project has been launched. The information systems of the members run well within the federation. During this project, another enterprises want to join this project with different expectation, such as different cooperation time periods, different cooperation domains, different expected results from federation, etc. As the result, to rebuild the federation is impossible. Our solution is to add one more federate, *WebservicesFederate*, as shown in figure 8. This federate will provide various services, different access permissions, and the common API of existing federation. The candidates can used the common API and the service they prefer to general their own federate, and connect to the existing federation with different authorities via the wide area network.

For example, in figure 8, there are two enterprise X and Y want to participate in the existing project. Enterprise X is a supplier and enterprise Y is a client who is interested in the final product of this project. Thus, enterprise X has to know the workflow which related to his business, and synchronize its information with other stakeholders. While, enterprise Y only ask the information from the federation, so, it doesn't have to synchronize with other systems. As the result, enterprise will ask *WebservicesFederate* for the service with an authority of synchronization with other federates, but enterprise Y will only require the service with the lowest authority, which can manage to listen to the federation. Finally, no matter what kinds of services they got, they are connected to the existing federation via web services.

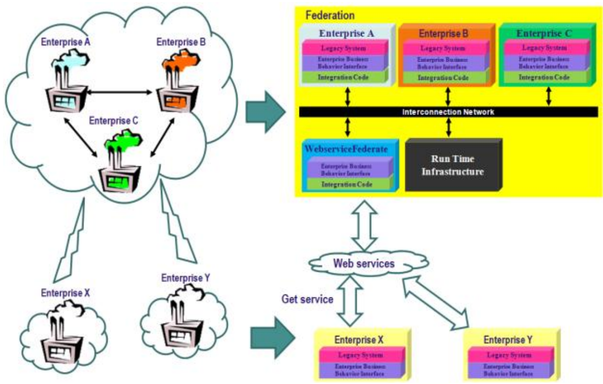

Figure 8 HLA Evolved Web Services

5.3.2. Architecture of HLA Evolved Web Services

The architecture of HLA Evolved Web Services is illustrated in figure 9. In our project, we choose an open source RTI, PORTICO (POTICO 2009), which doesn't provide Web-RTI functionality. Actually, PITCH RTI supports Web-RTI functionality (Möller 2007) , but it is not free, and as mentioned before, we want to initiate an open framework, so we choose PORTICO. So, we implement *WebservicesFederate* as a *bridge*, who takes charge of providing web services, connecting and synchronizing the federates outside the federation with the federates inside the federation.

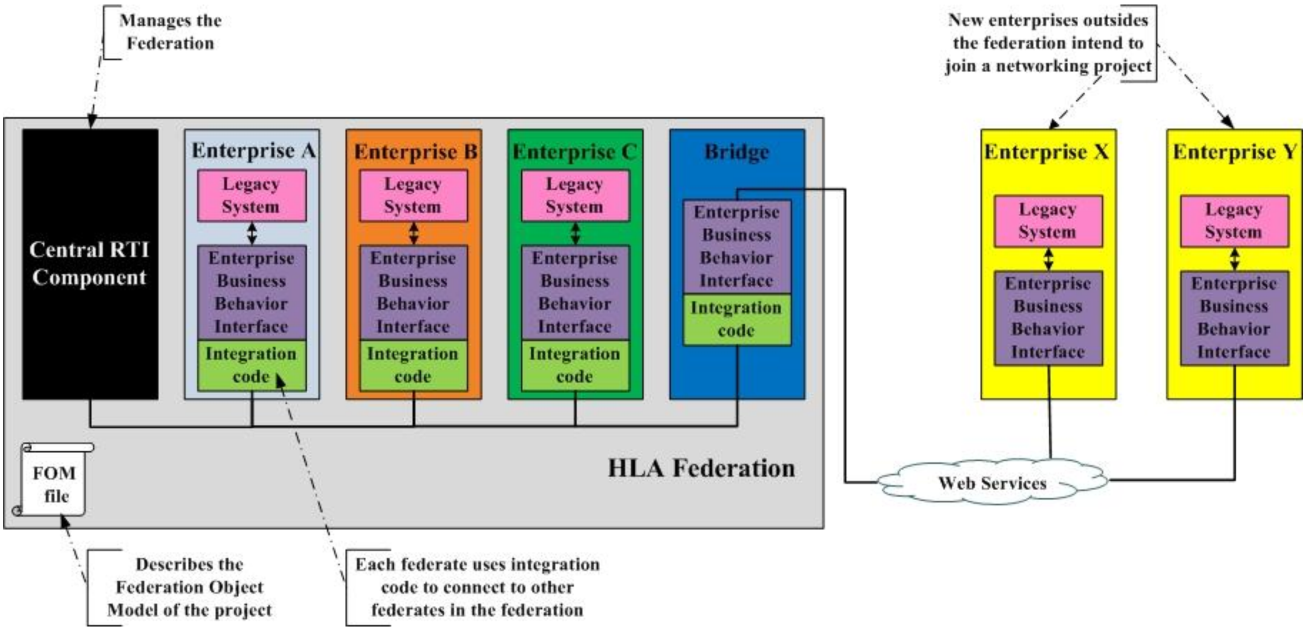

Figure 9 Architecture of HLA Evolved Web Services

As mentioned in section 5.1, after the harmonization of MDA and HLA FEDEP, we have *integration code* which provides a RTI independent API. And this API can be reused and published as common API. So, the candidates can reuse this API and follow the second scenario of model reversal, mentioned in section 5.2, to generate their own *Enterprise Business Behavior Interface* adapt to the common API. After that, the new federate outside the federation can send the information to *bridge* via the Web services interface and be synchronized by *bridge*.

5.3.3. Demonstration of HLA evolved web services

An implementation of the HLA evolved web services simulation has carried out by an internship student in our lab. It is based on POTICO RTI and Java. This simulation is implemented on Eclipse and can run in Windows NT or Unix system with JDK 1.6.0 (or higher) environment and ROTICO environment.

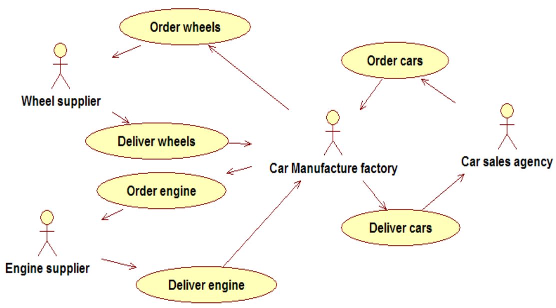

Figure 10 car manufacturing use case

This simulation describes a scenario of car manufacturing after a customer order arrives. This simulation simply simulates car manufacturer and suppliers of wheel and engine as figure 10 shows. Once the Car sales agency (*CSA*) receives an order from a client, it sends an order to the Car manufacture factory (*CMF*), then *CMF* sends orders to different suppliers, such as Wheel suppliers (*WS*) and Engine suppliers(*ES*), to get the parts or raw materials. When suppliers finish the products, they deliver them to the *CMF* where would assembles them and then deliver to the *CSA*. In this simulation, the status of the order is the concerned issue. *CSA* cares about when it can receive the cars it orders, and it concerns about the status of this order. The object class structure table shows in table 1.

Table 1: object class structure table

object class structure table		
Object root	WheelSupplier	dayToFinish
		currentState
		count
		price
	CarManufacturer	dayToFinish
		currentState
		count
		price
	EngineSupplier	dayToFinish
		currentState
		count
		price

We assume that *CSA* is the federate outside the federation, while, the *WS*, *ES* and *CMF* are the federates inside the federation. *CSA* is synchronized with *WS*, *ES* and *CMF* by *bridge*. Figure 11 illustrates the sequence diagram of this simulation. *CSA* requires the status of the order through web services, and *bridge* uses socket to connect the federation, and listen to the federation, waiting for the answer.

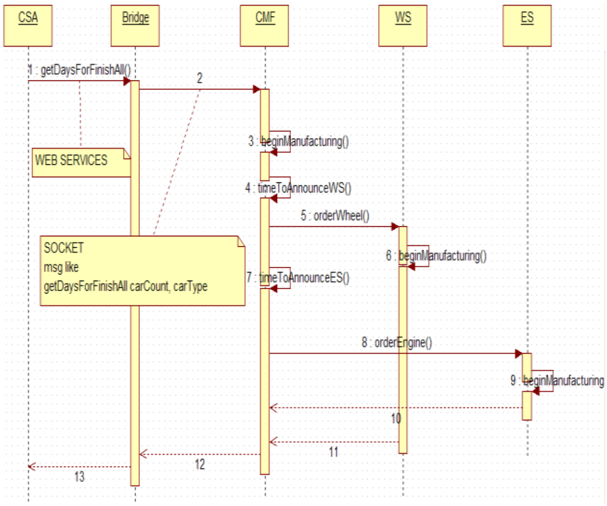

Figure 11 Sequence diagram of simulation

In this simulation, we assume that three seconds represent one manufacturing day. When *CSA* ask the probable manufacturing time for its order, it will receive a result, Gantt chart, as figure 12 shows. Then, when it requires the status of the order, it will receive the another Gantt chart with detail progress bar.

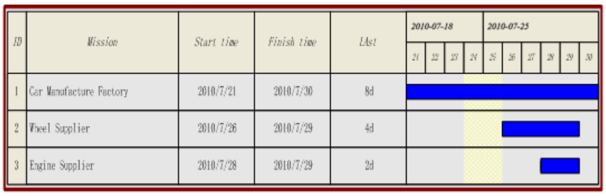

Figure 12 Simulation result

Now, the simulation runs well with laboratory data, except some gaps between our approach and the API provided by PORTICO. Recently, we are working on those gaps.

6. CONCLUSION

Based on the state-of-the-art, we have proposed a new systematic methodology, which is a valuable outcome of synthetic framework with harmonization of HLA FEDEP & MDA, Model reverse engineering and HLA Evolved Web Services. This methodology provides a new five steps process to develop models of simulation starting from conceptual enterprise models. In addition, it also bridges the gap from concepts to implementation in the field of enterprise modeling by offering a new standardized and reversible approach. And it also use the Web services to improve the performance of HLA. This methodology seems promising regarding to real enterprise information system requirement of distribution, federated interoperability and agility of adapt to dynamic context.

Up to now, this work is still in a research process. The methodology presented (each phase of HLA/MDA alignment and the model reversal process) still needs to be refined and detailed. An implementation of HLA Evolved Web Services has been carried out with an open source RTI, PORTICO. A case study will allow testing the proposed approach in an industrial context.

REFERENCES

Andreas T., 2002. Avoiding another Green Elephant — A Proposal for the Next Generation HLA based on the Model Driven Architecture. *Proceedings of the 2002 Fall Simulation Interoperability Workshop*, 02F-SIW-004.

Bourey, J.P., Grangel Seguer, R., Doumeingts, 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]

Richardson L., Ruby S., 2007, *RESTful web services*, O'Reilly Media, Inc. p. 299

Chen D., Vallespir B., Doumeingts G., 1997, GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology, *Computers in Industry*, Volume 33 Issue 2-3.

Doumeingts 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

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.

Favre L., Martinez L., Pereira C., 2008. MDA-Based Reverse Engineering of Object Oriented Code. *SERA'08*, 153-160.

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

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.

Jean B., 2005. On the Unification Power of Models. *Software and Systems Modeling*, volume 4, number 2, pp 171-188

Jouault F., Jean B., Mikaël B., 2009. Towards an advanced model-driven engineering toolbox. *Innovations in Systems and Software Engineering*, volume 5, pp 5-12.

Möller B., Clarence D., Mikael K., 2007, Developing Web Centric Federates and Federations using the HLA Evolved Web Services API. *Proceedings of 2007 Spring Simulation Interoperability Workshop*, 07S-SIW-107.

Möller B, Löf S., 2005, Mixing Service Oriented and High Level Architectures in Support of the GIG. *Proceedings of the 2005 Spring Simulation Interoperability Workshop*, 05S-SIW-064.

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 [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].

POTICO, 2009, *Developer Documentation*. Available from: http://porticoproject.org/index.php?title=-Developer_Documentation [accessed 15 July 2010]

Shawn P., 2003. The Next Step Applying the Model Driven Architecture to HLA. *Proceedings of the 2003 Spring. Workshop*, 03S-SIW -123.

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.

Trbovich S., Reading R., 2005. Simulation and Software Development for Capabilities Based Warfare: An Analysis of Harmonized Systems Engineering Processes. *Proceedings Spring Simulation Interoperability Workshop*, 05S-SIW-106.

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).

Ullberg J., Chen D., Johnson P., 2007. Barriers Driven Methodology For Enterprise Interoperability. *IFIP International Federation for Information Processing*, 453-460.

Zacharewicz G., Chen D., Vallespir B., 2008. HLA Supported Federation Oriented Enterprise Interoperability, Application to Aerospace Enterprises. *Proceeedings 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)

AUTHORS BIOGRAPHY

Zhiying TU is PHD student of University Bordeaux 1. His research subject is modelling and simulation of interoperable distribute enterprises. He recently focused on distribute simulation based on HLA and model reverse engineering.

Gregory ZACHAREWICZ is Associate Professor in Bordeaux 1 University (IUT MP). His research interests include Discrete Event Modelling (e.g. DEVS), Distributed Simulation, Distributed Synchronization Algorithms, HLA and Workflow. He recently focused on Enterprise Modeling and Interoperability. He has published more than 20 papers in Conference and International Journals. He is Reviewer in Conferences (Summer SCS, WinterSim...) and SCS Simulation journals.

David CHEN is Professor at University Bordeaux 1. His research interests focus on enterprise modelling and interoperability. He has been actively participating in EU program (FP1-FP6), several cooperation programs between EU and China. David CHEN is member of CEN TC 310/WG1 and ISO TC184/SC5/WG1 (Modelling and architecture) and involved in IFIP WG5.12 and IFAC WG5.3 (Enterprise Integration and Networking). He has published more than 80 papers in international journals and conferences.

Zhenchuan Zhang is Master student of University Bordeaux 1. Recently, he is doing his internship in IMS/LAPS. The subject of his internship is "Distributed enterprise information system interoperability based on HLA evolved SOA".