MULTI AGENT/HLA ENTERPRISE INTEROPERABILITY (SHORT-LIVED ONTOLOGY BASED)

G. Zacharewicz(a), O. Labarthe(b), D. Chen(c), B. Vallespir(d)

[(a)(c)(d)] IMS-LAPS / GRAI, Université de Bordeaux – CNRS, 351, cours de la Libération, 33405 TALENCE cedex, France
[(b)] CIRRELT, Université Laval, 2325 rue de la Terrasse, bureau 2535, Québec city, Québec, Canada

[(a)]gregory.zacharewicz@ims-bordeaux.fr, [(b)]olivier.labarthe@cirrelt.ca, [(c)]david.chen@ims-bordeaux.fr, [(d)]bruno.vallespir@ims-bordeaux.fr

ABSTRACT

This paper aims at proposing an implementation of the Federation oriented Enterprise Interoperability concept, using Multi Agent / HLA paradigm and the rising notion of Short-Lived Ontology. We give first, a review of ongoing researches on Enterprise Interoperability. Then, we recall on Artificial Agent Concept and HLA Standard that appear to be adequate to support execution of the studied concept. Indeed, on the one hand Agent dialogue fits the concept of information exchange in a federated enterprise interoperability approach, on the other hand the HLA standard, initially designed for military M&S purpose, can be transposed for enterprise interoperability at the implementation level, reusing the years of experiences in distributed systems. From these postulates, we propose the first Agent/HLA framework Short-Lived Ontology based to implement distributed enterprise models from the conceptual level of federated enterprise interoperability approach.

Keywords: Enterprise Interoperability, Multi-Agent-Systems, HLA, Ontology

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 skill 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 semantics necessary to assure interoperability (IDEAS 02).

This paper proposes a new contribution of Information Technology (IT) architecture and platform to implement Enterprise Interoperability. In the first part, we present the various approaches of interoperability and the current consideration of interoperability stated as conclusion of the Interop Network of Excellence (FP6, 508011) (Chen 07). Then, we recall the concepts of software Agent and the High Level Architecture (HLA), i.e. a standard for distributed simulation.

Next, from our experience, we propose to investigate three aspects of interoperability. The first concerns time management in Enterprise Interoperability; the dynamic aspect has to be tackled with sound techniques. The second aspect concerns the definition of Enterprise Ontologies; federated approach in interoperability requires a new definition of high-level standard (i.e. Ontology) for exchanged data. The last aspect concerns the privacy of data. Indeed, enterprises must manage confidentiality of data shared between entities; levels of rights on enterprise data must be defined.

Computer science Ontologies, Artificial Agent language, and Object/Interaction in HLA can give keys to two first considerations. As well, the experience coming from Information Systems (IS) and M&S programming can be studied to keep data safe to address third point. From these postulates, we specify a platform implementation using HLA and Software Agents' autonomous dialogue concepts, to the concern of distributed federated Enterprise Interoperability models.

2. BASIC CONCEPT OF INTEROPERABILITY

Enterprise Interoperability refers to the ability of interactions between enterprise systems. The interoperability is considered as significant if the interactions can take place at least at the three different levels: data, services and process, with a semantics defined in a given business context (IDEAS 02).

Interoperability extends beyond the boundaries of any single system, and involves at least two entities. Consequently establishing interoperability means to relate two systems together and remove incompatibilities. Incompatibility is the fundamental concept of interoperability. It is the obstacle to establish seamless interoperation. The concept 'incompatibility' has a broad sense and is not only limited to 'technical' aspect as usually considered in software engineering, but also ‘information’ and ‘organization’, and concerns all levels of the enterprise (Chen 07).

Our goal is to tackle interoperability problems through the identification of barriers (incompatibilities) which prevent interoperability to happen. Basic concepts relating to Enterprise Interoperability are categorized into three main dimensions as described below.

2.1. Dimension of interoperability barriers

This dimension takes into account three categories of interoperability problems.

Conceptual barriers are related to the problems of syntactic and semantic of information to be exchanged. This category of barriers concerns the modeling at high levels of abstraction as well as the level of coding.

Organizational barriers are related to the definition of responsibilities and authority so that interoperability can take place under good conditions.

Technological barriers are related to the problem of use of information technologies. This category of barriers concerns the standards that are used to present, store, exchange, process, and communicate data through the use of computers.

2.2. Dimension of interoperability concerns

This dimension identifies various levels of enterprise where interoperability takes place. These levels are based on ATHENA Architecture.

The business level refers to working in a harmonized way at the levels of organization and company in spite of for example, the different modes of decision-making, methods of work, legislations, culture of the company and commercial approaches etc. so that business can be developed and shared between companies.

The process level aims at making various processes working together. A process defines a sequence of services according to a specific need of a considered company. Commonly, in a company, several processes run in interactions (serial or parallel). In the case of a networked enterprise, internal processes of two companies must be connected to create a common process.

The service level is concerned with identifying, composing, and making function together with various applications (designed and implemented independently) by solving the syntactic and semantic differences, as well as finding connections to various heterogeneous databases. The term `service' is not limited to computer-based applications but manual ones as well.

The data level refers to making different data models (hierarchical, relational, etc.) and different query languages working together. The interoperability of data is related to find and share information coming from heterogeneous bases, which can moreover reside on different machines with different operating systems and databases management systems.

2.3. Dimension of interoperability approaches

This dimension takes into consideration the three admitted approaches to develop interoperability as illustrated in figure 1.

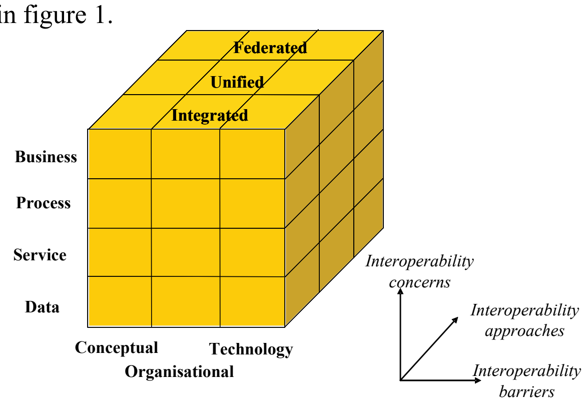

Figure 1: Framework for enterprise interoperability

Integrated approach: there exists a common format for all models. This format must be as detail as models. The common format is not necessarily a standard but must be agreed by all parties to elaborate models and build systems.

Unified approach: there exists a common format but only at a meta-level. This meta-model is not an executable entity as it is in the integrated approach but provides a mean for semantic equivalence to allow mapping between models.

Federated approach: there is no common format. To establish interoperability, parties must accommodate on the fly. Using federated approach implies that no partner imposes their models, languages and methods of work.

Today, most of the approaches developed are unified ones such as for example in the domain of enterprise modeling, we can mention UEML (Unified Enterprise Modeling Language UEML (2003) and PSL (Process Specification Language) which aim at supporting the interoperability between enterprise models and tools.

Using the federated approach to develop Enterprise Interoperability is most challenging and few activities have been performed in this direction. The federated approach aims to develop full interoperability and is particularly suitable for an inter-organizational environment (such as networked enterprises, virtual enterprises, etc.). In the Enterprise Interoperability roadmap published by the European Commission (IST 06), developing federated approach for interoperability is considered as a research challenge for the years to come.

3. HLA RECALLS

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.

3.1. HLA concepts

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 00).

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 00).

In order to respect the temporal causality relations in the execution of distributed computerized applications; HLA proposes to use classical conservative or optimistic synchronization mechanisms (Fujimoto 00).

3.2. HLA Implementation Components

An HLA federation is composed of federates and a Run Time Infrastructure (RTI) (IEEE 2000), figure 2.

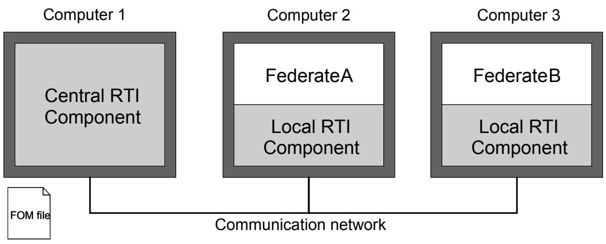

Figure 2: HLA Component Framework

A federate is a HLA compliant program, the code of that federate keeps its original features but must be extended by other functions to communicate with other members of the federation. These functions, contained in the HLA specified class code of *FederateAmbassador*, make interpretable by a local process the information received resulting from the federation. Therefore, the federate program code must inherit of *FederateAmbassador* to complete abstract methods defined in this class used to receive information from the RTI.

The RTI supplies services required by distributed executions, it routes messages exchanged between federates. It is composed of two parts.

The "Local RTI Components code" (LRC, e.g. in figure 2) supplies external features to the federate for using RTI call back services such as the handle of objects and the time management. The implementation is the class *RTIAmbassador*, this class transforms the data coming from the federate in an intelligible format for the federation. The federate program calls the functions of *RTIAmbassador* to send data to the federation or to ask information to the RTI. Each LRC contains two queues, a FIFO queue and a time stamp queue to store data before delivering to the federate.

Finally, the "Central RTI Component" (CRC, e.g. in figure 2) manages the federation notably by using the information supplied by the Federation Object Model (FOM) (IEEE 2000) to define Objects and Interactions classes participating in the federation.

A federate can, through the services proposed by the RTI, "*Publish*" and "*Subscribe*" to a class of shared data. "*Publish*" allows diffusing the creation of object instances and the update of the attributes of these instances. "*Subscribe*" is the intention of a federate to reflect attributes of certain classes published by other federates.

4. AGENT-BASED DISTRIBUTED SIMULATIONS

The Multi-Agents System (MAS) concentrates on the study of the collective behavior which results from the organization and interactions of agents for the resolution of problems. A MAS is a distributed system in which there is generally no centralized control or global point of view. A MAS is composed of agents which act in an autonomous way but do not locally have the knowledge, the resources or the information required to ensure the coherence of the concerted actions in a MAS. This section is dedicated to the presentation of the agent definition and the introduction of distributed simulations of Agent-Based Systems.

4.1. Agent Definition

Actually there is no consensus in the scientific literature on the definition of an agent. Disciplines in which reference is made are numerous and various authors have proposed different definitions as for example Ferber (1999) and Huang and Nof (2000). However, the definition proposed in Jennings *et al.* (1998) is commonly used within the MAS community: "*an agent is a computer system, situated in some environment that is capable of flexible autonomous action in order to meet its design objectives...*». In Wooldridge *et al.* (1995) the authors define the concept of an agent according to the following properties:

autonomy: an agent operates (task selection, decision-making, etc.) without human or other direct intervention and neither the actions it realizes nor its internal state are submitted to any intervention;

reactivity: an agent perceives its environment and reacts in an appropriate way to the environment changes;

pro-activeness: agents are able to act by taking initiatives driven by their goals;

social ability: agents are able to interact with other agents through communication language or social rules.

The importance accorded to the properties expressed above depends on the application needs and nature. Agents and MAS constitute an active research field in which numerous applications are developed. In Nwana (1996) and Wooldridge (2002) the authors propose surveys of the agents according to various application domains (cognitive agents, software agents, mobile agents, etc.). Agents perceive the modifications of their environment and perform actions on it. Among the possible actions, agents have to determine the most suited decisions that can reach their objectives. In addition to the application domain, environment, interaction and organization influence the design of the agent.

4.2. Distributed simulations of Agent-Based Systems

The properties which characterize MAS are particularly adapted for the modeling and simulation of distributed and dynamic systems. For the distributed simulation of multi-agent systems several tools and methodologies are defined in the literature. The following sub-section is dedicated to the presentation of research works focused on the integration of multi-agent systems through a HLA federation.

Lees *et al.* (2007) presented a tool named HLA_AGENT which integrates the SimAgent toolkit and the HLA. The SimAgent toolkit is used for the development of alternative agent architectures which can evolve in various environments. Concerning the organization of HLA_AGENT, a federate is associated to a SimAgent which is in relation with a Federate Ambassador (handles callbacks from the RTI) and a RTI ambassador (handles calls to the RTI).

Song *et al.* (2007) proposed a multi-agent data collection system to improve the performance of existing mechanism in HLA simulation systems. Each agent is connected to the RTI and is composed of four functional modules: Communication, Cooperative Decision, Data Logging and Data Processing. Furthermore, each agent is responsible of a Database composed of a subset of the simulation data.

Minson and Theodoropoulos (2008) introduced a middleware layer named HLA_REPAST to facilitate the integration between HLA and the sequential MAS simulation toolkit Repast. The authors supplied a complete description of the necessary steps associated to the creation of a federation of interacting instances of RePast models within HLA.

Chen *et al.* (2008) was interested of the data accessing problem to optimize the execution of distributed simulation of agent-based systems. In this context two routing approaches was proposed. The first named range-based attribute locate data according to a set of attribute value range, and the address-based approach locate a particular state variable from a given identifier. In order to study the dynamic of the two approaches the authors used the PDES-MAS framework proposed in Logan and Theodoropoulos (2001).

Cicirelli *et al.* (2009) proposed to interface a set of agents (grouped under the name of Theatre) to the HLA/RTI via two components. The TransportLayer component is connected to an RTI Ambassador for transmission and reception of messages. The ControlMachine component is connected to a Theater Ambassador for the management of the Theatre logical time and the external received simulation messages.

5. PERSPECTIVES TO INTEROPERABILITY

5.1. Removing Barriers to Interoperability

From the state-of-the-art of federated enterprise interoperability and implementations experiences presented in § 2, we can define several directions for, almost natural, interoperability barrier removal with Agent and HLA concepts in the following domains.

The first direction concerns the definition of commonly recognized paradigms and data structure able to evolve during run time.

The second not addressed requirement at the enterprise modeling level is the data synchronization. The data exchanged order is crucial; ignoring this can lead to not desired indeterminist model behavior.

Finally the enterprise modeling must consider the confidentiality management of data. The interoperability can be considered between concurrent enterprises in that context, a strategy of data sharing/not sharing between these must be defined. We present, in the following, propositions to address these requirements.

5.2. Enterprise Model Transformation Methodology for Distributed Execution

From the postulate that different enterprise models implementation using HLA, each of them follows its own development cycle. (Zacharewicz *et al* 09) introduced a common methodology by converging HLA FEDEP (IEEE 2000), MDA (OMG 2003) and MDI (Bourey 2007) steps, to clarify and rationalize the implementation method and the models. This life cycle proposed to standardize the steps to reach simulation or implementation from a conceptual enterprise model. This formalization will help reuse of development knowledge and will give a common metric to compare solution developments.

Phase 1: In first step, the objectives of the federation of enterprises have to be defined. Basically, the common goal of all federation created by this methodology consists in defining federation of interoperating enterprise models. This representation can naturally use the typical enterprise relation model elaborated in the enterprise model interoperability field (e.g. UEML models (2003)).

Phase 2: As second step, the mapping of enterprise models into HLA federates is realized. In detail, the way models handle received information and how they send information to the federation is addressed, these mechanisms can conform to synchronization algorithm proposed in (Zacharewicz 2008a). We pay here attention to reuse already existing enterprise models. Not existing enterprise models federates are created. In addition, we define information to exchange, i.e. what is the structure of distributed ontology and messages.

Phase 3: In the third step, the methodology maps enterprise interoperating connections between models into HLA interactions and objects. Then, these data are structured to generate the associated FOM. The strategy concerning confidentiality of data is also explicitly addressed in that step. Besides, to respect time constraints, objects and interactions among enterprise federates are time stamped related to local logical time of supplier enterprise federate to be handled and delivered right in time by RTI.

Phase 4: The federation is executed. The results obtained by simulation are used for validation of the models by test and analyze. In case it does not fulfill the specification, the methodology allows feedback correction as described in the FEDEP last step.

6. AGENT FEDERATED ENTERPRISE INTEROPERABILITY

6.1. Short-lived ontology concept

In the federated Enterprise Interoperability approach, no common persistent ontology is supposed to exist; the communication must be accommodated on-the-fly. In consequence the ontology that structures the messages exchanged must be short-lived, (i.e. non persistent). We state that the communication mechanism, in this approach, can be informally illustrated as follow in figure 3. We mainly distinguish two cases.

In case a., the enterprise 1 sends information and the ontology to decode it at the same time. This ontology is supposed to be only valid for this information.

In the case b., the enterprise 1 sends only the information to enterprise 2. Once enterprise 2 receives the information, it checks within 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 new received ontology can be conserved to be reused with further data sent by the same emitter.

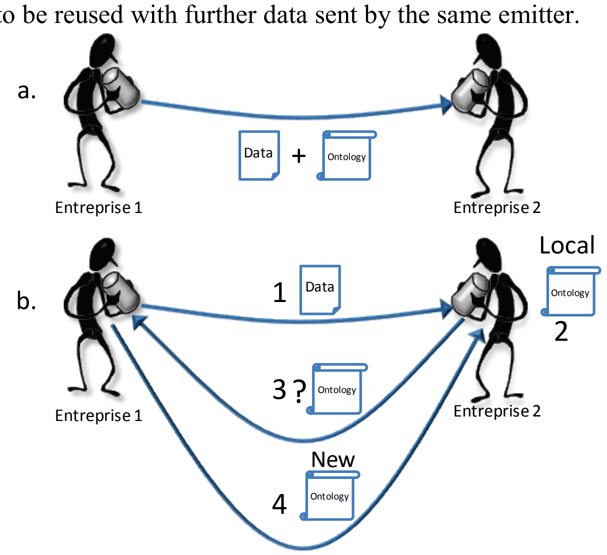

Figure 3: Federated Interoperability Data Exchange

In the first solution, the information size exchanged is more important, it can be intercepted and the confidentiality can be broken. In the second case the confidentiality is enforced but it can require more exchanges between the two partners and consequently overlapping the communication duration. Nevertheless, for confidentiality (i.e. §4.3) and accuracy to the definition of §2.3, we choose to focus in this article, on the second solution. From that postulate we introduce the concept of "short lived" ontology (our ontology definition can be based on Gruber (1995) definition), where ontology can be, in some case, suppressed after use or have finite duration validity. It maps the on the fly accommodation requirement of federated interoperability.

6.2. Agent for Short-lived ontology concept

From the concept presented in the preceding point, we state that the autonomous dialogue between Agents, from Multi Agent System (MAS) (Ferber 95) and Agent Based Simulation (Huang and Nof 2000), can map properly the "on the fly" concept of federated interoperability at process level. We propose to use the dialogue mechanism algorithm of Agent programming, introduced in (Ferber 1995) and (Huang and Nof 2000), to solve at computerized level the problem of federated Enterprise Interoperability. This dialogue between Agents, aims at establishing communication (e.g. two enterprises that discuss to agree on domain ontology), it is based on Agents cooperation behavior settings, and messages exchanged language (ontology).

On the one hand, (Ferber 1995) specifies the communication behavior algorithms of Agents' with Petri Nets (PN). On the other hand, (Zacharewicz *et al* 2008a), have tailored the use of DEVS/HLA (introduced by (Zeigler *et al* 2000) as a generalized M&S language (including PN) for distributed systems, gaining accuracy and flexibility (these models communicate within a distributed environment by message passing). Thus, we state that DEVS/HLA synchronized communication can support a unified, reusable and interoperating implementation of distributed Agents' dialogue.

On behalf of previous paragraphs propositions, we propose to develop a MAS simulator in the aim of validating Enterprise Interoperability concepts, studying the performance by simulation and implementing a concrete solution for Enterprise Interoperability ISs.

The research for developing MAS distributed Platforms is wide (as can denote for e.g. a repository of Agent-Based Simulation Platforms proposed in (Marietto *et al* 2002)), and actual MAS simulators are powerful (i.e. (Huang and Nof 2000)). Nevertheless, they mainly do not tackle the problem of interoperability and reuse of components at coding level (e.g. heterogeneity of syntax, semantic, time management, etc.). In consequence, to preserve Interoperability at all levels of Enterprise Modeling including execution level; we propose to implement an Enterprises Federated Agents System that will be HLA compliant (to guaranty also run time interoperability between heterogeneous software components).

7. IMPLEMENTATION OF FEDERATED INTEROPERABILITY COMMUNICATION

7.1. Framework Definition

The proposition starts from the statement on interoperability needs on interfacing enterprises IS in the context of project cooperation. The figure 4 depicts the requirement on exchanging data from heterogeneous Information Systems, including vendor tools such as SAP and other specific developed solutions. It is issued by generalizing study case of (Zacharewicz 2008a), various enterprises are involved in a common project and their heterogeneous components need to be interfaced. Existing interoperability between components is represented with plain arrows and in demand interoperability with dotted arrows. (Labarthe 2007) reports on solution to establish interoperability using MAS in the communication of enterprises IS (i.e. figure 4 long dotted set); they have implemented an agent communication mechanism using JADE platform facilities. Zacharewicz *et al.* (2008a) defined an interoperable Workflow using DEVS (Zeigler 2000) / HLA (i.e. figure 4 short dotted set). By joining these approaches, this paper introduces the basis for a generalized approach to realize interoperability between heterogeneous components. Two ways of research are envisaged.

The first requirement is solving the interoperability of data and services, HLA can be part of a solution.

For instance in the practical case of figure 4, a solution is to establish links to an "Interoperability Service Bus", referring to Enterprise Service Bus of Chapel (2004) concepts, to connect new features with already connected components, (e.g. DEVS/HLA Anylogic, HLA…). We detail in the next point how an HLA compliant platform can facilitate the integration of all required components.

The second point is trying to take into account the requirement of interoperability at each development process steps and how it can facilitate the interoperability at lower level of abstraction. This idea in MDI (Bourey 2007) is to define interoperability models at the Business level of modeling enterprise and to facilitate process level interoperability, and then to develop data interoperability services coherent to previous levels of abstraction. This point is not detailed in this paper but is still in our scope of studies (Zacharewicz et al. 2008b).

7.2. Definition of Ontology Level

Different levels of ontology are required in our approach. From low level with poor semantics associated to HLA objects to information transport level (HLA bus). Agent KQML (Finin 1994) will be used as an intermediate level able to match from low level description to high level description used in heterogeneous platforms, software or enterprise models involved in the system using reference to domain or application Ontologies. For instance, one challenge is to be able to transform and transport SAP, Anylogic or ARENA descriptions of a problem through a communication in a distributed network.

7.3. HLA Compliant Execution Model

As enounced in previous section, this new interoperability concept of Enterprise Federated Agent needs to be tackled at run time. Based on our experience in HLA support, we propose an innovative implementation of Enterprise Interoperability Federation (i.e. figure 5).

The "Interoperability" components layers can be added to IS of enterprise either they are ad hoc developed or vendor solutions. The idea is to add a component to code and decode information exchanged with the original IS, this component is considered as black box and no modification is realised on it. We present in detail in this section the components required for this global distributed platform.

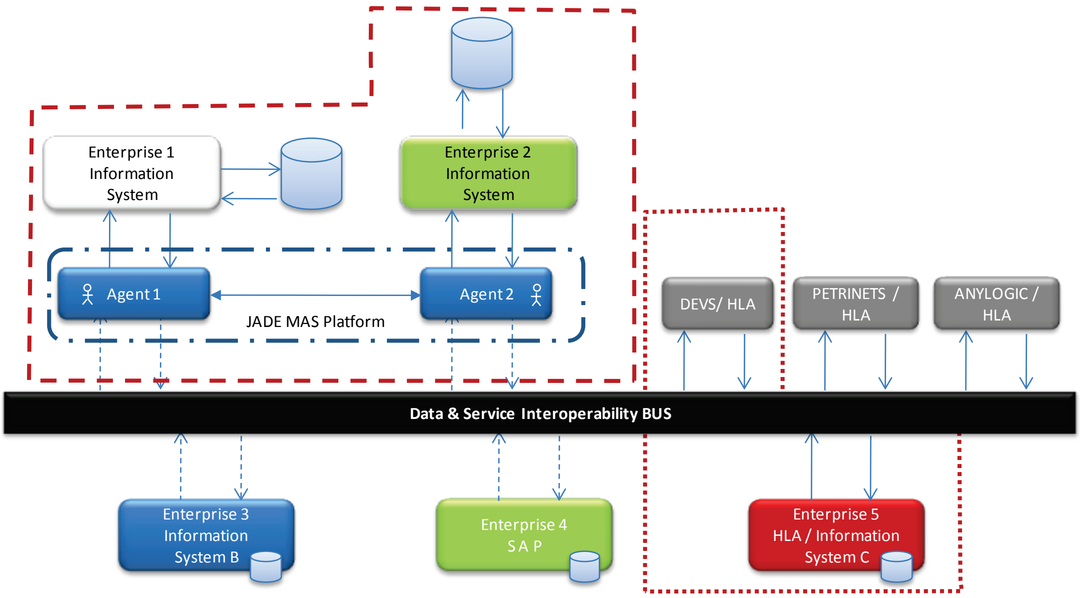

Figure 4: Federated Interoperability Data Exchange

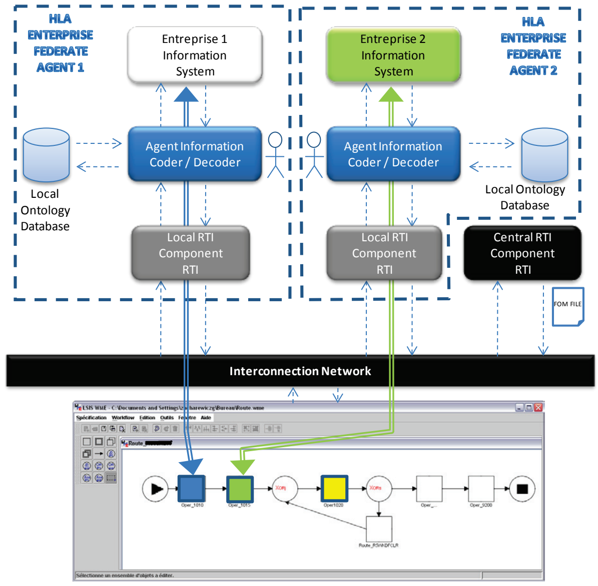

Figure 5: Federate Agent Data Exchange Process flow

7.3.1. Information System Services Layer

The distributed implementation requires the extension of two add-ons to the local enterprise IS to define HLA Enterprise Federate Agent (HEFA). We present in the figure 5, the elements of this new architecture. The respective local enterprise ISs remain unchanged, HLA only required to add components to interface with input output messages of the IS.

7.3.2. Multi Agent Coding / Decoding Layer

All agents involved in the data ontology matching are detailed:

Agent 1 Storing data: This agent is employed to store the received information and will check the capacity to decode information using a communication with ontology agent. Receiving the agreement to use the data, it sends data and ontology to the information system

Agent 2 Ontology: This agent is linked to a repository of local ontology; it checks the consistency of the information regarding the local ontology and decides if the data can be exploited. If yes, it sends back to the storing data agent the information and the ontology to use. If not, it asks to the communication agent to start dialog to obtain the appropriate ontology.

Agent 3 Communication: This agent will start a conversation with the respective agent of the data sender to deal on the modality to receive the appropriate ontology. We propose, in the following, that this dialog will be established using HLA message communication protocol but from a conceptual consideration it can be considered as a general approach where HLA is just one practical technical solution.

7.3.3. Local RTI Component Layer

This level is the lower level; it is the service and data level. It deals with technologies employed to exchange computerized information.

The first component is the Local RTI component (LRC), i.e. §3.2. This code is required to insure the reception and the emission of information within the transposing from the local description of data to the HLA format. This component manages also the time by delivering the message according to their timestamp and to the local time.

The second one is the Agent Information Coder Decoder (ICD). This new specific software component, introduced in this paper, will be able to analyze the information received by querying the local ontology data base to determine the capacity to interpret the received message according to the behavior introduced in § 6.1. If the local ontology is not able to interpret the received message, ICD will ask through the LRC for the associated ontology to the sender of the message. At the end, this is the technical transcription of the Agent communication behavior.

We illustrate in figure 5, the use of two instances of our structure connected to a Workflow monitoring tool. This tool is employed to run a simulation of a Supply chain or of a document exchanging process and is triggering right in time the information systems of the interoperating enterprises.

7.4. HLA Interaction and Object Class Model

The idea is to propose a new mapping for data to be exchanged in a HLA compliant distributed system between Interoperable Enterprise Agents. We propose in figure 4, a generic HLA FOM that will support the descriptions of the data required to insure the exchange of information in the figure 3. case b.

HLA object Root (N)	Information channel (S)	Information channel 1 (PS)
		Information channel 2 (PS)
		Information channel 3 (PS)
		Information channel 4 (PS)
		Information channel ... (PS)
		Information channel n (PS)
HLA Interaction Root (N)	Ontology (S)	Ontology 1 (PS)
		Ontology 2 (PS)
		Ontology 3 (PS)
		Ontology 4 (PS)
		Ontology … (PS)
		Ontology n (PS)

Figure 6: Federate Enterprise Interoperability FOM

Information to exchange: Information exchanged between enterprises information systems will be mapped with HLA Object class models (that handle persistent information in the distributed execution). The enterprise IS federates will publish and subscribe (PS) to these classes of information. An information channel Object (i.e. figure 6) represents the informational link between at least two enterprises. We notice that communication channels preserve confidentiality.

Ontology: Ontologies exchanged between enterprises are not persistent in the studied approach; they will be mapped into HLA Interactions (that are non persistent information exchanged). Enterprise IS Federate

Agents will publish and subscribe (PS) to these classes of information. One Ontology class (i.e. figure 6) is associated to each information channel; each Ontology definition can change during run time. Eventually, validity duration can be set for each Ontology.

The structure of the generic ontology and of the messages will be implemented in the FOM presented in figure 6. Because of generic concepts introduced in this paper, HLA interaction parameters and HLA object attributes are not fully specified; they will be in more detailed depending on the applications. It gives flexibility to the data structures exchanged.

7.5. Data Exchange Time Management

To respect causality, we recall in that section, an HLA time management specification of exchanged messages sequence between HEFAs introduced in (Zacharewicz 2009).

At first point, each HEFA sets its channels of communication that link it with other HEFA it is interested to interoperate. For that purpose, it defines Publish/Subscribe (PS) participation to HLA objects defined in the FOM (see figure 6). At the same time, the HEFA can set the ontology's it is able to provide (P) and the ones it is interested to receive (S) at the beginning; (this status will evolve during run time because of creation and destruction of non persistent ontology).

In figure 7, we describe the time management mechanism to exchange information between two interoperating HEFAs. The services and call backs provided by the RTI are here mentioned by directly referring to HLA 1516 standard reference book (IEEE 2000).

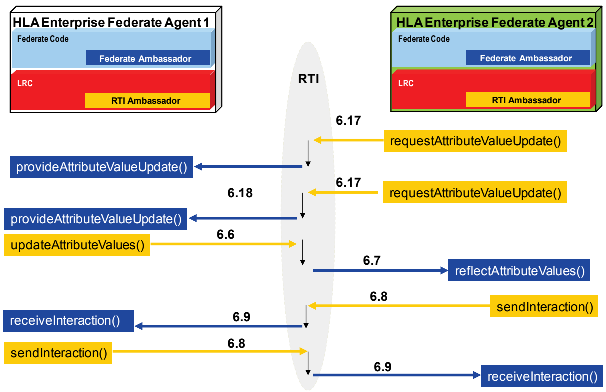

Figure 7: Federate Data Exchange Time Management

Assuming an HEFA2 is interested in information emitted by HEFA1. HEFA2 must subscribe to the information published by an HEFA1. Then, the steps of information exchange during run time are described in figure 7. In first step, HEFA 2 asks for information from HEFA1 by calling *requestAttibuteValueUpdate()* RTI service (e.g. in figure 7, HEFA 2 asks for two data to HEFA 1). The RTI transmits these demands to EF1 with *provideAttibuteValueUpdate()* callback. When HEFA1 is ready to distribute these information or their new values (regarding to time constraints) it will deliver them to HEFA2 with *updateAttibuteValues()*. HEFA2 possesses now the information. It transmits it to its information coder / decoder component (i.e. figure 4). This component checks the capacity to decode the message with its local ontology. If it is not able, it asks for the associated ontology of the message with a non persistent questioning to HEFA1, i.e. by using the service *sendInteraction()*. We notice that these actions are transcript from Agent behaviour specification. Once HEFA1 receive*Interaction()* call back, It will supply the ontology with the same communication process (*sendInteraction()* to HEFA2). In fine, the ontology is just sent and received; no common semantic information persists between HEFA1 and HEFA2. Yet, the ontology received can be locally stored to simplify and speedup next data exchange between HEFA1/HEFA2.

8. CONCLUSION AND FUTURE WORK

This article has given a state of the art of Enterprise Interoperability concepts and illustrated the use of Agent concepts and HLA standard for the implementation of enterprise applications federations.

From the new concept of short-lived enterprise ontology for federated Enterprise Interoperability, we proposed a specific implementation of distributed enterprise models for simulation or real time information exchange. At the end, the keys for implementation given by Agent dialogue mechanism has helped to bridge the gap from Enterprise Interoperability concepts to HLA compliant distributed implementation in the field of Enterprise Modeling by following a new standardized and systematic approach.

Currently, the environment is at specification and conception time. Meantime of verification and validation of our approach, the development of a beta version is beginning in the aim to test, in particular, the implementation of the short-lived Ontologies. The next step of detailed conception and implementation will be initiated in the perspective of the rising partnership between IMS Bordeaux and CIRRELT Québec labs.

9. REFERENCES

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 in 15 June 2009]

Chappell D. A., 2004. *Enterprise Service Bus*, Portland: O'Reilly.

Chen, D., Dassisti, M., and Elvesæter, B., 2007. *Enterprise Interoperability Framework and Knowledge Corpus - Final report Annex: Knowledge Pieces*, Contract no.: IST508 011, Deliverable DI.3 Annex, Available from: http://www.interop-vlab.eu/ [accessed in 15 June 2009]

Chen, D., Ewald, R., Theodoropoulos, G. K., Minson, R., Oguara, T., Lees, M., Logan, B. and Uhrmacher, A. M., 2008. Data access in distributed simulations of multi-agent systems. *Journal of Systems and Software*, 81(12), 2345-2360.

Cicirelli, F., Furfaro, A. and Nigro, L., 2009. An Agent Infrastructure over HLA for Distributed Simulation of Reconfigurable Systems and its Application to UAV Coordination. *Simulation*, 85(1), 17-32.

Ferber, J., 1999. *Multi-agents Systems. An Introduction to Distributed Artificial Intelligence*, London: Addison Wesley.

Finin, T., Fritzson, R., McKay, D., and McEntire, R., 1994. KQML as an agent communication language. *Proceedings of the Third international Conference on information and Knowledge Management*, November 29 - December 02, Gaithersburg, (Maryland, USA).

Fujimoto, R.M., 2000. *Parallel and Distributed Simulation System*, New York: Wiley Interscience.

Huang, C. Y. and Nof S. Y., 2000. Autonomy and viability measures for Agent-based manufacturing systems. *International Journal of Production Research*, 38(17), 4129–4148.

IDEAS Consortium, 2002. *Thematic Network, IDEAS Interoperability Development for Enterprise Application and Software Roadmaps,* Annex 1 – Description of work. Available from: http://www.interop-vlab.eu/ [accessed 15 June 2009]

IST. 2006. *Enterprise Interoperability Research Roadmap*, Final Version, July, Available from: http://cordis.europa.eu/home_en.html [accessed 15 September 2007]

IEEE std 1516.22000, 2000. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification*, New York: Institute of Electrical and Electronic Engineers.

Jennings, N.R., Sycara, K. and Wooldridge, M., 1998. A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-agents Systems,* 1(1), 7-38.

Labarthe O., Espinasse, B., Ferrarini, A. and Montreuil B. 2007. Toward a Methodological Framework for Agent-Based Modelling and Simulation of Supply Chains in a Mass Customization Context. *Simulation Modelling Practice and Theory*, 15(2), 113–136.

Lees, M., Logan, B. and Theodoropoulos, G. K., 2007. Distributed simulation of agent-based systems with HLA. *ACM Transactions on Modeling and Computer Simulation*, 17(1).

Logan, B. and Theodoropoulos, G. K., 2001. The distributed simulation of multi-agent systems. *Proceedings of the IEEE - Special Issue on Agent-oriented Software Approaches in Distributed Modelling and Simulation,* 89(2), 174–185.

Marietto, M. B., David, N., Sichman J.S. and Coelho, H., 2002. Requirements Analysis and Agent-Based Simulation Platforms: State of the Art and New Prospects. *Lecture Notes in Artificial Intelligence,* Berlin: Springer-Verlag, 2581, 89–104.

Minson, R. and Theodoropoulos, G. K., 2008. Distributing RePast agent-based simulations with HLA. *Concurrency and Computation: Practice & Experience*, 20(10), 1225-1256.

Nwana, H.S., 1996. Software agents: An overview. *Knowledge Engineering Review*, 11(3), 205–244.

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]

Gruber T. R., 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing, In: *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Dordrecht: Kluwer Academic Publishers.

Wooldridge M., 2002. *An Introduction to Multi-agents Systems*, NY: John Wiley & Sons.

Song, H. J., Shen, Z.Q., Miao, C.Y., Tan, A.H. and Zhao G. P., 2007. The Multi-Agent Data Collection in HLA-Based Simulation System. *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, pp 61-69, June 12-15, San Diego, (CA, USA).

UEML, 2003. *Public Web for UEML*, EU Project IST–2001–34229, Available from http://www.uemlwiki.org [accessed 15 June 2009]

Wooldridge, M. and Jennings N.R., 1995. Intelligent Agent: Theory and practice. *Knowledge Engineering Review*, 10(2), 115–142.

Zacharewicz, G., Frydman, C., and Giambiasi, N., 2008a. G-DEVS/HLA Environment for Distributed Simulations of Workflows. *Simulation*, 84(5), 197–213.

Zacharewicz, G., Chen, D., Vallespir, B., 2008b. HLA Supported, Federation Oriented Enterprise Interoperability, Application to Aerospace Enterprises. 08E-SIW-074. *Proceeedings of 2008 International Simulation Multiconference EuroSISO,* 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. , April 22-23, Beijing (China)

Zeigler, B., Kim, T.G., Praehofer, H., 2000. *Theory of Modeling and Simulation*, New York: Academic Press.

AUTHORS BIOGRAPHY

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.

Olivier LABARTHE is Associate Professor of operations and decisions systems in the Faculty of Administrative Sciences at Laval University in Québec, Canada. He received a PhD in Administrative Sciences and Computer Science, from Laval University and Aix-Marseille III University respectively. His research interests focused on the multi-agent modeling and simulation for the design and the management of manufacturing and logistics networks.

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.

Bruno VALLESPIR is Professor at University Bordeaux I and member of IMS. Within this framework, he is today responsible of the research group about production systems. Bruno VALLESPIR's research topics are production management, performance evaluation and enterprise modelling including modelling tools and design methodologies. On these topics, he published more than 60 papers in conferences, journals and books, was involved in around 20 international, European (from FP1 to FP6, mainly ESPRIT) or French projects and working groups. Recently, he was involved in INTEROP NOE. He is member of several working groups and steering committees of IFIP and IFAC.