ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

MODELING OF MEDIATION SYSTEM FOR ENTERPRISE SYSTEMS COLLABORATION THROUGH MDA AND SOA APPROACHES

SAIB SARA, BENMOUSSA RACHID, BENGOUD KENZA

Information logistic and production systems Team, National School of applied sciences (ENSA), Cadi Ayyad Univerity-Marrakech, Abdelkarim Khatabi Avenue, BP 575, Morocco

E-mail: saibsara@gmail.com, benmoussa.ensa@gmail.com, kbengoud@gmail.com

ABSTRACT

The interoperability between enterprise systems and their integration deeply depends on their ability to collaborate efficiently. Therefore, in this paper we study the collaboration of enterprise systems through the analysis of collaboration of business processes. We propose to design a collaborative architecture based on the MDA principles. The CIM level of the approach targets the construction of the collaborative business process (CBP). To design a CBP that responds efficiently to the need, it is necessary to develop a knowledge based system to generate the CBP, the KBS is composed of three main parts: the first one concerns the acquisition of participant knowledge, in the second part we define two ontologies, collaborative ontology to structure knowledge about the future collaborative network, and business process ontology to structure captured knowledge for future use. A set of deduction rules is defined to extract knowledge from ontologies and the last part focuses on the composition of the collaborative business process using BPMN language according to extracted information. In the PIM level we define a set of transformation rules that transform the collaborative business process generated by the KBS into a set of services based on SOA principles. In this paper, we will present the theoretical aspects of each part of the approach as well as the tools used in each part.

Keywords: *Interoperability, Collaborative Business Process, Ontology, MDA, SOA, BPMN.*

1. INTRODUCTION

The impact of the globalization phenomenon facing the companies for the last couple of years is best illustrated in the amount of increasing competition. In order to be and remain competitive in a market where the customer has become even more demanding in terms of time, cost and quality, companies must react quickly and accommodate the specific needs of their customers. Only by doing so will they acquire a leading position in these new emerging markets.

All major groups in an industry, including companies, rely on software to ensure the execution of their activities. For example, ERP (Enterprise Resource Planning) software integrates business processes and optimizes all functions of organization and its management such as project management, inventory management and purchases, general and analytical compatibility, etc [1]. New systems of technical management such as MES (Manufacturing Execution) can monitor and manage tools of production and collect real-time data pertinent to the production process in order to control and assure product quality [2]. Other needs have also been discovered in the field of personalization and information about products, which has led to emergence of systems that manage technical documentation such as PLM (Product Life Cycle Management), which allows technical capitalization of all of a company's product information [3]. These systems are intended to complement each other and form a structure of open transmission and sharing of information, but often a single system ends up executing tasks that are not normally assigned to it. Implicitly, the integration between these components poses the problems of sharing responsibilities and functions between these systems and the methods in which information must be exchanged.

Information system is continuously adapted to changing business practices and needs, and software should support this changing. As a consequence, information systems are becoming more complex and the software architecture of the enterprise becomes completely disorganized, and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

all that negatively influences the management strategy of the company. The most appropriate solution to this problem is to make business processes in the core of the solution. The aim is to design and control the organizational structures in a very flexible way, so they can easily adapt to a new environment.

Our research work aims to solve the problem of interoperability between systems. Our study of the state of the art revealed that collaboration between systems is a necessity within companies but difficult to achieve. Firstly, it is the case because the enterprise systems exhibit a lack of flexibility, secondly the majority of software are proprietary and are not designed to cooperate with other programs and finally because of the incompatibility in the information representation and in the software application methods adopted [4]-[5]. Indeed, the collaboration between heterogeneous systems requires the development of a well-defined framework that allows all software to connect their various processes within a global one.

Our main aim is to interconnect heterogeneous systems regardless of their technologies and to provide the necessary tools to predict collaboration scenario and make it as dynamic as possible. Indeed this work is divided into two complementary problems. The first one is the identification of business processes and their collaboration in order to have a global one, the second one is the migration to service oriented architecture by transforming the collaborative business process into a set of services following guidelines of service oriented approach. To define the system development life cycle we intend to follow Model-Driven Engineering (MDE) principles.

Service-oriented architecture (SOA) is an approach to loosely coupled, protocol independent, standards-based distributed computing where software resources available on the network are considered as Services [6]. SOA provides a service-based approach for interoperable systems development and integration. So, it offers mechanisms of flexibility and interoperability that allow different technologies to be dynamically integrated, independently of the system's platform in use.

The Object Management Group (OMG) has been proposing the model driven architecture (MDA) as a reference to achieve wide interoperability of enterprise models and software applications [7]. The Model Driven Architecture (MDA) reduces significantly the gap between the needs of business and their implementation. The MDA is based on model transformation, so the system construction consists of a sequence of models and transformations between these models therefore its main advantage is the ability to transform a CIM to PIM and one PIM into several PSMs.

Integrating SOA with MDA gives the opportunity to bring the services definition to a higher level of abstraction, adding agility, flexibility, due to the more formal and accurate platform-independent specification of the services requirements and design.

In this paper, we propose a service oriented Model-Driven approach to collaborate business process. These business processes are expressed using modeling notation language (BPMN) since it seems to be the appropriate language that offers constructs to represent most optimally business activities, operations and constraints.

According to [8]-[9], the CIM is a model of a system that shows the system in the environment where it will operate. It helps in understanding a problem and defining a shared vocabulary for use in the models of the other levels. Therefore, to answer this need, we propose the tools to collect adapt and transform different kinds of knowledge about the collaboration with the aim to deduce the CIM requirements in order to smartly automate the design of the collaborative business process. The collaborative business process has to be able to follow business process evolutions and know how to adapt to a new context.

The first level of the approach targets the development of a knowledge-based system to provide the necessary knowledge to design a well-defined business process able to adapt to any context. In the second phase we define a set of mapping rules that transform BPMN model represented as a CIM model to Service model developed at the PIM level and let us simply consider here that PSM has been conceived with an Enterprise Service Bus (ESB) as an ultimate target platform [10].

The paper is structured as follow: section 2 discusses some related works; section 3 presents the proposed approach. The rest of the paper is structured in the same way as levels of the approach. Thus, section 4 presents the steps of the design of the business process that is based mainly on the development of knowledge-based system. Section 5 describes the transformation process of collaborative process model to SOA models. Finally, section 7 concludes and presents the perspectives and limits of this paper.

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. RELATED WORKS

Numerous scientific research studies have been conducted for many years to examine the interoperability concept. Several solutions have been proposed, but the issue of interoperability is still a hot topic due not only to its non-stop emerging constraints, but also to the many changes that arise daily. The interoperability problem arises at three levels: data, resources and business processes. Several research works define frameworks to characterize levels of interoperability: European Interoperability Framework (EIF) [11], ATHENA Interoperability Framework (AIF) [12], Interoperability Development for Enterprise Applications and Software (IDEAS) [13], and e-Government Interoperability Frame-work (e-GIF) [14].

There are several definitions for interoperability which vary based on the functional use and the systems involved. Interoperability is the ability of two or more systems or components to exchange and use information [15].

The interoperability of systems is not only a technological or conceptual problem, but it may also be an organizational problem. Indeed, according to the EIF [11], there are three types of interoperability.

- Technological interoperability is used for presenting, storing and exchanging data via computer hardware.
- Semantic interoperability ensures that the information exchanged is understood in terms of its meaning and its interpretation by systems using it.
- Organizational interoperability defines responsibilities, authorities and organizational structures necessary for the acceptance of the exchange of information between applications by different actors.

In a distributed and heterogeneous environment, software components of information systems of the enterprise must collaborate to meet business and technical needs expressed by different collaborators. The configuration of these components is not only technological but also organizational and conceptual.

There are two possible architectures to achieve interoperability between several information systems. The first one is the point-to-point architecture. It is the one in which each system is connected to another system through an interface. The second one, called the mediation architecture, is based on an entity called the mediator of information that allows the collaboration and communication between several applications [16]. The mediator architecture is more agile than the first architecture and easier to integrate. The current proposed solutions achieving the interoperability of systems are based on predetermined and automatic links and physical connectors that are based on specific standards (message formats, connectors, etc.). They create tightly coupled systems that are based on monolithic architectures, in other words they cannot interconnect systems if the development technologies aren't the same. They also require synchronous operations: all systems involved must work in the same time frame of reference, and when the number of systems to interconnect increases, interoperability becomes difficult and sometimes impossible [17]. The architecture based on the mediator is the best solution to solve the problem of interoperability because it allows service coordination and exchanging of information between different information systems in a transparent manner.

Reference [18] defines the three main functions for interoperability as shown below in Figure-1.

- Conversion and provision of data
- Management of systems
- Orchestration of Collaborative Process

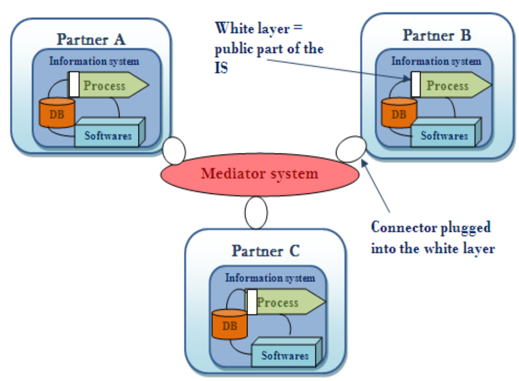

Figure1: Information Mediator System

The public layer of each system and the mediator system both are the support components of the interoperability.

The mediator covers both informational and operational exchanges. It is a tool of translation and linking data, applications and processes of partners. It has to orchestrate the collaborative process and manage syntactic and semantic correspondence of data and applications partners The three centralized functions introduced in the conceptual definition of the Mediator given above can therefore be implemented in the following terms:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- An ontology based knowledge management of data
- A partner's service registry
- A workflow management system to support collaborative process executions

Starting from this high level definition of the Mediator, collaboration is designed and developed at the business level.

As we explained in the previous section, our issue is divided into two parts, the first one focuses on the business level of interoperability and how to model the collaborative process while the second one targets the transformation of the collaborative process to a set of services. These two issues have been studied in several works and different architectures and methodologies that have been proposed to solve the issue.

Reference [10] proposes to generate a collaborative information system through the transformation of models. It proposes an approach based on MDA that transforms the collaborative business process represented with BPMN into a collaborative information system expressed on UML language. This work presumes that the involved organizations are able to propose the collaborative process model. Hence, collaboration rules are not clearly defined in the CIM level. The solution of this problem is to provide a knowledge-based system that defines the collaborative process from the network and participants information.

Reference [19] completes the approach proposed by [10] by addressing the business level of the interoperability, and the CIM of the approach, so it deals with the collaboration between enterprises through their business process. A knowledge-based system is developed to supporte the design of collaborative processes. These works tackle the interoperability inter-enterprise; they propose approaches to define virtual enterprise through the definition of inter-enterprise collaborative process.

Reference [19] designs the collaborative process such as a chain of abstract services. Reference [20] proposes an architecture that transforms abstract services into concrete services that are able to adapt to the context of collaboration and use inheritance relationship to construct a common process. The majority of works around the interoperability are based on SOA and MDA architectures because they guarantee a high level of flexibility and the ability to reuse.

The World Wide Web Consortium (W3C) refers to the service-oriented architecture (SOA) as "a set of components which can be invoked, and whose interface descriptions can be published and discovered" [21].

The SOA is an approach for the development of enterprise systems by loosely coupling interoperable services. It offers an effective solution to the problems of reusability, interoperability and it reduces coupling between applications. Therefore it allows the creation of systems from a set of business services without taking into account technologies used during the development of these applications. The goal is to break down the functionality of an enterprise into a set of functions encapsulated in a component that will communicate with its environment using messages. The web services are considered the most important implementation of the SOA model. They are channels to interoperate with different applications, regardless of platform or environment in which they are executed [22].

Model driven architecture (MDA) has been proposed as an approach to deal with complex software systems. The MDA is a software design approach proposed by the OMG (Object Management Group) [23] to support the development of complex and distributed systems. This is a particular variant of the model driven engineering (MDE) [8]. Models are at the heart of the MDA approach. So, its principle is to split the development process into three separate model layers and automatically transform models from one layer into the other. Thus, this architecture is designed to incite interoperability of the information models independently of the framework in use (operating system, modeling and programming language, data servers and repositories) [5].

The MDA comprises three main layers [8]-[9]:

- The Computation Independent Model (CIM) layer indicates the more abstract models. According to (OMG 2003), these models focus on the environment and requirements of the system and describe concepts of a given domain.
- The Platform Independent Model (PIM) layer describes all or part of the software functionalities and behavior without worrying about technical details.
- The Platform Specific Model (PSM) layer is associated with a specific platform based on a well-defined technology

In our research we adopt this engineering approach in order to define a model of the mediator system, starting from a set of business processes. Our first contribution will be the collaboration

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

between two business processes and addressing the transition between business and logical levels in the mediator development life cycle. We are interested in this work to generate a PIM system model mediator based on services oriented architecture. This model can then be projected on several technology platforms (ESB, EAI, etc.).

3. SUGGESTED APPROACH

The aim of the proposed approach is to model the MIS in order to define a collaborative process based on services.

The methodology is composed of two parts developed separately but they are complementary and dedicated to reach the same goal. The first part defines the modeling of collaborative processes through a study and analysis of the collaborative environment. The second part deals with the design of CIS (collaborative information system) through the proposition of a mechanism that transforms the collaborative process model into a system based on services.

The methodology adopted in this work is based on the identification of the collaborative system using a set of transformation rules of meta-models and models. It allows starting from a business specification (a set of business models) to find at the end the model-oriented services. It follows instructions of the MDA (Model driven architecture). The MDA is used to describe separately the parts of a software system independent of specific platforms and implementation.

The language we will use to represent our collaborative processes is BPMN because it is dedicated to business process modeling and it covers both information and organizational aspects of process.

The approach is composed of several phases that are presented below in the figure-2. The development process is an iterative process allowing incremental development. The rest of the paper is structured in the same way as the steps of the development process. The phases are:

- Step 1: The KBS (knowledge base system) modeling and the construction of collaborative process according to the collaborative knowledge and reasoning provided by the KBS.
- Step 2: The transformation rules between the collaborative process and the SOA business.
- Step 3: The implementation

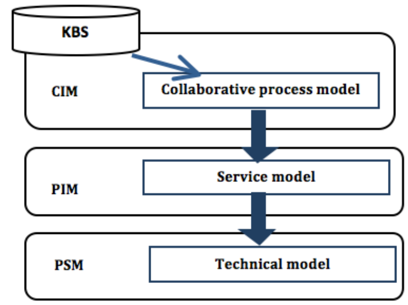

Figure 2: The suggested approach

Figure 3 shows the technical architecture of the prototype, it represents the development process steps, tools and technologies that will be used in each phase to implement the system functionalities. The system is composed of six parts:

- Knowledge acquisition to define the collaboration network.
- Deduction of the collaboration model according to the reasoning done about the knowledge gathered from the network. This step is the main one in the KBS life cycle since it uses ontology approach and requires a high level of precision and expertise.
- Definition of collaborative process according to specific knowledge extracted from the knowledge based system.
- Construction of the BPMN process.
- Transformation of BPMN process to a set of services
- System Implementation

Figure 3: Functionalities of the prototype and development technologies

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. THE MODELING OF KNOWLEDGE BASED SYSTEM AND DEFINITION OF COLLABORATIVE PROCESS

According to [8], the CIM is a model of a system that shows the system in the environment where it will operate. It helps in understanding the specified domain and the case to model in order to collect and define a common vocabulary that will be used in the models of others levels. In our case the CIM targets the process collaboration, so the knowledge has to come from the process taking part in the network, responsibilities, business and objectives. Therefore our aim is to collect structure and transform the knowledge about the collaboration and processes with the aim to provide necessary information to model the collaborative process taking into account the requirement of CIM level.

A knowledge-based system has been developed in order to store, use and generate the key knowledge to produce an increasing collaborative process. The KBS is very useful for analysts when this case of collaboration is often repeated and evolved, so it has to be stored for future use, or when expert is not available, or when intelligent assistance and training are required to take a strategic decision or to solve a collaborative problem in the most flexible ways.

The KBS development process is composed of three parts:

- Knowledge acquisition
- Knowledge verifications and validations
- Knowledge representation

4.1 Knowledge Acquisition

Knowledge acquisition is the process of extracting, structuring and organizing knowledge; it is one of the most important phases in the development of knowledge-based system. Therefore, it has to be carefully planned. It deals with extracting knowledge from sources of expertise to collect information about the environment of collaboration and the partners involved in this collaboration. Knowledge about collaboration can be acquired from many different sources in many different ways such as conversation, interview, software documentation, experiences, etc. The quality of sources and quantity of information influences the performance of the knowledge base system. A good quality and a large amount of data lead to accurate characterization of collaboration so that the generated collaboration scenario will be closer to the need.

The knowledge acquisition focuses on collecting necessary -knowledge related to collaboration. According to [24]-[25] collaboration has an individual and collective aspect. The individual aspects concern the actors who accomplish the collaboration tasks. The collective aspect concerns the strategies, goals, relationships, as well as processes.

According to [24] there are several parameters for configuring collaborative networks such as partners, common goals, duration stability, relationships between partners and the organizational structure which explain how partners are connected to each other.

To cover all collaboration concepts, it is necessary to gather all knowledge about processes involved in the collaboration such as the aim of the process, the operations and functions performed in the making or treatment of a product or service, resources used to accomplish a task. These concepts will be accurately defined in the next section.

If domains are defined and knowledge are well extracted, the collaboration project chances for success will be greatly increased.

The aim of this first part is to present knowledge in a well-defined format understandable and usable by the system. To facilitate the knowledge gathering, it is advisory to present the users with graphic tools that would enable him to represent his knowledge adequately. So, it is possible to develop a graphic tool to design the collaborative network using technologies such as GMF (Graphical Modeling FrameWork). This is not the heart of our contribution, therefore we use XML to represent and organize the knowledge; the XML file obtained is the collaborative network model. This file will be imported to the knowledge-based system. The domain concepts that will be represented in this step have to be in accordance with those modeled in the KBS. The collaboration model will mainly represent the concepts of the ontologies that will be defined in the next section. At this level it's necessary to interview the system users in order to identify and extract the objectives and concepts of collaboration.

4.2 The Reasoning

To define a common vocabulary and a common understanding of information in a domain, and to share information between people and software the ontology seems to be an adequate mean to represent the concepts and relations between them. Ontology is an explicit formal specification of the terms and relations between them in a domain of interest [26]. Therefore, it provides a well-defined knowledge structure that can be used and reused in order to facilitate the

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comprehension of concepts and relationships in a given domain, and the communication between different domain actors. Ontology gives also the necessary knowledge that axioms use to predict the system behavior and its response due to an event.

In many works related to business process collaboration, the involved components are considered as predefined, and are not supposed to be modified when they participate in any collaboration. So, the first issue that this paper deals with is how to collect knowledge and information about partners and the collaborative environment to design and predict the appropriate collaboration. Ontologies appear to be the best method to structure and present collaborative network and business process knowledge.

To capture knowledge about collaboration domain, different ontologies have been developed but each one represent a specific domain of collaboration, as result it doesn't exist any common collaboration ontology which can be used to model process descriptions for different kinds of collaboration [27].

An approach for an ontology-based process is given by [24]. It defines a collaborative network ontology composed of collaboration ontology and collaboration process ontology. The network collaboration ontology used in their work represents the mainly concepts of inter-enterprise collaboration. The aim is to design a virtual enterprise via the collaboration of enterprise business process. In our research work we deal with the enterprise systems business processes collaboration. Therefore, we propose a first ontology that represents concepts linked to collaborative network of enterprise systems as shown in figure-4. The second one illustrates the business process that presents the various aspects of industrial process as shown in figure-5.

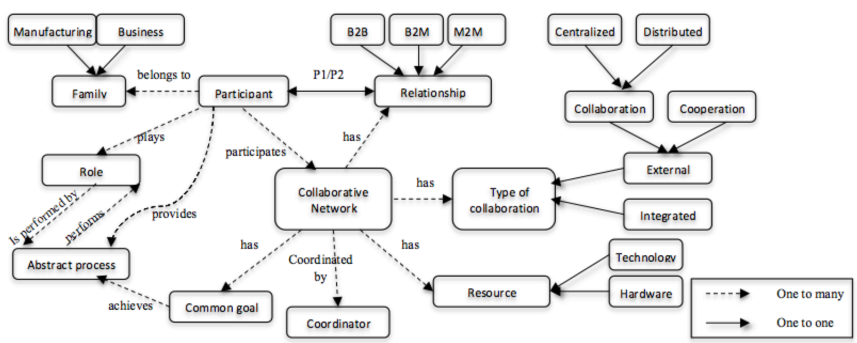

Figure 4: Collaborative network ontology

The concepts of Collaborative network ontology are:

Collaborative network is a group of at least two participants who work together to achieve one or multiple common goals.

Common goal describes reason why the network is established in terms of products or services to sell or achieve for the consumer or in response to a need inside the enterprise.

There are two kinds of collaboration the first one is the integrated collaboration and the second one is the external collaboration. In the first one, the business processes involved in the collaboration are implemented and executed in the same software. In external collaboration, business processes belong to different and heterogeneous systems. The external collaboration is divided into two types. The first one is the cooperation. In this case, all participants are involved in the execution of the collaborative business process, which is supervised by all involved participants. Yet, no common goal is set and each one has its own purpose to achieve separately. The second type is the collaboration. In this situation, all participants collaborate to achieve the same goal. The collaboration can be centralized [28]-[29]; the collaborative process is composed of business processes activities of participants, and only those activities are executed according to the collaboration rules managed by the coordinator. The collaboration can also be distributed, in this case each participant executes its process and the exchange of data occurs by sending messages.

The participant is an enterprise system stands in a complementary position with the other systems the enterprise software architecture. Those systems are composed of a set of business processes. This system belongs to a business or a manufacturing family. Manufacturing systems are those that operate in the industrial production of goods for use or sale. Business systems are those linked to the management of resources, customer relationships, supply chain etc.

The abstract process is a high-level process that describes the aim of a process for

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

example: purchase process, sale process, and procurement process.

A role defines the responsibility of the participant in the network.

Relationship defines the interaction between two participants. It describes the three types of relationships that are Business to business, Business to manufacturing and manufacturing-to-manufacturing.

A Coordinator is a software unit responsible of managing and controlling the collaboration.

Resources are means used to achieve the collaboration. They can be a set of equipment or technologies.

In this approach we deal with the business process. Hence, in this part we study the industrial characteristics of the process in order to propose a business Process Ontology that conceptualize both collaborative concepts and industrial characteristics of the process. An ontology called Process handbook has been developed at the MIT Center for Coordination Science and it provides a specialization of processes and their inter-relationships [30]. An other ontology called CPO ontology is suggested in [24] to represent de collaborative business based on component that will support the collaboration such as MIS service, follows of resources shared between services, etc. Other metamodels have been suggested to represent the business processes [31]-[32]. These works integrate both static element such as role, actor and resources, and dynamic elements such as events and coordinator and represents relations between concepts. The suggested ontology gathers depicted concepts and adds other.

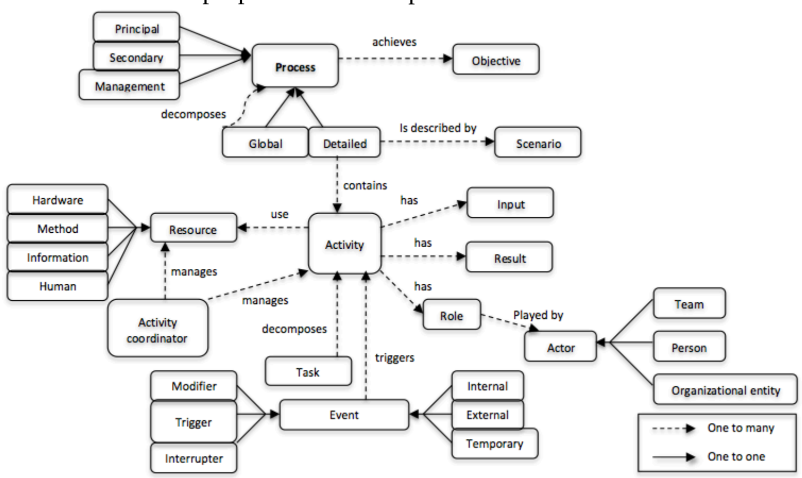

Figure 5: Business process ontology

The concepts that make up the business process ontology are:

A Process: A process is a dynamic system that executes a set of instructions to achieve a goal.

The objective of a process: the objective of the process is the expression of its purpose, for example: the purchase of a product, the management of human resources.

Global process: It represents the goal only, and it can be decomposed in processes.

Detailed process: it describes the contents of the system showing the details of its operation mainly its activities.

Principal process: its objective reflects the principal purpose of the system at a high level to which it belongs.

Secondary process: its contribution is not strategic.

Management process: its aim is the control of other processes.

The scenario: a text or a diagram to simulate the execution of an instance of the process, it is usually fictitious.

Activity: activities describe how the goal of a detailed process can be achieved. It is a set of operations to be performed by machines and / or humans. This set may belong to production, communication or control operations.

Actor: an active element (individual, an organizational entity or a machine) responsible for one or more activities in a process

A role: a set of activities performed by a single actor.

A task: the smallest element of an activity, the task has no autonomy compared to the activity that it depends on.

The event: It is a stimulus that causes a reaction in an activity. The event can be divided

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

into three categories: internal, external, or temporary events.

An internal event is a stimulus that is generated within the boundaries. They may be the boundaries of process, domain boundaries within which the process is executed or the boundaries of the organization.

A temporary event: has one or periodical term (time, frequency or deadline elapsed) which is associated with a response of the organization.

An external event is a stimulus that we can not manage. It comes from an actor or external system.

The effect caused by an event is the triggering of the corresponding activity. However, in some cases the event type may be taken into account during the execution of the activity. This is what leads to a second specialization of the event according to his purpose.

A trigger event induces the execution of the first task of the activity. This is the most frequent case.

An interrupter event leads to stop the activity even if all the tasks have not been completed. In this case others specific tasks can be performed.

A modifier event changes the sequence of the process.

The result is a product of the execution of an activity. A result may be a resource, an input or an internal event to another process activity.

An input is an element that will be changed during the execution of the activity.

The resource is an element used for the execution of an activity.

A condition expresses a restriction on the execution of a task, or on the transition triggering.

To design the ontologies, we use Protégé it is a free, open source ontology editor and knowledge base framwork [33].

The reasoning level is an important phase in the life cycle of the knowledge-based system because at this level the system analyzes the collaborative situation through the study of stakeholders information, their performance area and their relationships.

At this level, it is necessary to define rules to ensure the morphing of the collaboration knowledge into the business process knowledge. In other words these rules have to be defined to connect the two sub-ontologies included in CNO. They are as well applied to the collaboration model to propose a model of collaborative process. The deduction rules are defined based on some references found in the literature.

We opted for SWRL Semantic Web Rule Language (SWRL) [34] to write deduction rules. It is designed to be used in the context OWL-DL and, thereby, inherit important semantic characteristics that make automated reasoning more flexible [23].

We specify five rules that we explain in this section.

GR1: As shown before each enterprise rely on a set of systems which compose the software architecture, and if we apply the definition given in the SWRL Specification [34]-[35]-[24] to our context and consider each system as an independent entity, we can say that inside the software architecture each system is defined by its goals, the abstract processes to achieve these goals and the role that perform the abstract processes. So there is a relation of dependency between role and abstract process.

According to this analysis two rules are defined. The first one derives the role when the abstract process is recognized, while the second one derives the abstract service when the role is recognized.

Participant(?x) ∧ playRole(?x, ?y) ∧ performAProcess(?y, ?z) → provideAProcess (?x, ?z)

This rule gives as a result the list of abstract processes that can be performed according to a given role the participant plays.

GR2:

Participant(?x) ∧ provideAProcess(?x,?y) ∧ isPerformedByRole(?y, ?z)→playRole(?x, ?z)

This rule gives as a result the list of roles played by a participant when the abstract process is recognized.

GR3: When a participant provides an abstract process, and this abstract process is associated with a process then, the participant provides the process. This rule exposes the list of processes that the participant has to present.

Participant(?x) ∧ provideAbstractProcess(?x, ?y) ∧ hasProcess(?y, ?a) → provideProcess (?x, ?a)

GR4: When families of the two participants are known we can deduce the type of relationship that link them.

Cnetwork(?a) ∧ hasrelationship(?a, ?z) ∧ P1(?z, ?x) ∧ belongToFamily(?x, ?b) ∧ P2(?z, ?y) ∧ belongToFamily(?y, ?b), differentfrom(?a, ?b) → hasRelationship(?a, B2M)

GR5: This rule infers the MISservice that manages the collaboration between two participants, which provides two different processes.

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CNetwork(?a) ∧ hasRelationship(?a, ?z) ∧ P1(?z, ?x) ∧ provideprocess (?x, ?b) ∧ P2(?z, ?y) ∧ provideprocess(?y, ?c) → iscordinatedby(?a,?f) ∧ MISservice(?f)

The proposed rules have not been completed and several rules can be defined from the two ontologies.

This step was dedicated to the design of the KB and to the automation of the deductive reasoning. At this level, we deduce the collaborative process model according to the reasoning made on the collaborative model obtained in the first step. To import the collaboration model into the knowledge base, it is necessary to transform it into OWL model. In this case it is a transformation from an XML to XML. It can be simply made by XSLT stylesheet. The KB is developed on the basis of the CNO.

We use OWL to represent the ontology. As shown in the previous section to represent the deductive rules we use SWRL. There are several tools proposed in the market for editing OWL and SWRL, but the most suitable and persistent one is protégé framework. It is a free, open source ontology editor and knowledge base framework developed by the Research Department of Medical Informatics School of Medicine, University of Stanford (USA). Protégé and OWL are based on the same components that are Classes (concepts), Properties, Individuals (instances) and Rules.

To write the deduction rules, we use the SWRL editor. To execute SWRL rules, it is necessary to use an inference engine. The inference engine is the brain of an expert system it is a program that derive answers from a knowledge base. Several open source inference engines exist in the market such as SweetRules, Jena, Pellet, OpenRules, Jess. The only inference engine that work with SWRL Editor is the Jess engine.

The aim of the next section is to represent the collaborative process using BPMN language, so we will have to transform the XML file to a BPMN diagram.

5. THE CONSTRUCTION OF THE COLLABORATIVE PROCESS

The objective of this section is the modeling of collaborative process using BPMN language. To transform XML file to a BPMN Process, it is necessary to define the transformation rules from XML language to BPMN language.

Firstly, it is compulsory to describe and define the concepts and gateways according to which BPMN is based.

In the BPMN, a process is divided into one or more Pools. A pool represents the major participants in a process, typically separating different organizations. A pool contains one or more lanes. A lane is used to represent involved actors and to organize and categories activities within a pool according to their function or role. The process basic components are either the Activity that can be a task if it is elementary or sub process when it is composed, or events. These components are interconnected by sequence flows within a pool and shows message flows exchanged between pools. Gateways are used to model control flow branching in BPMN. Gateways split and join sequence flow. Figure 6 shows relations between these concepts.

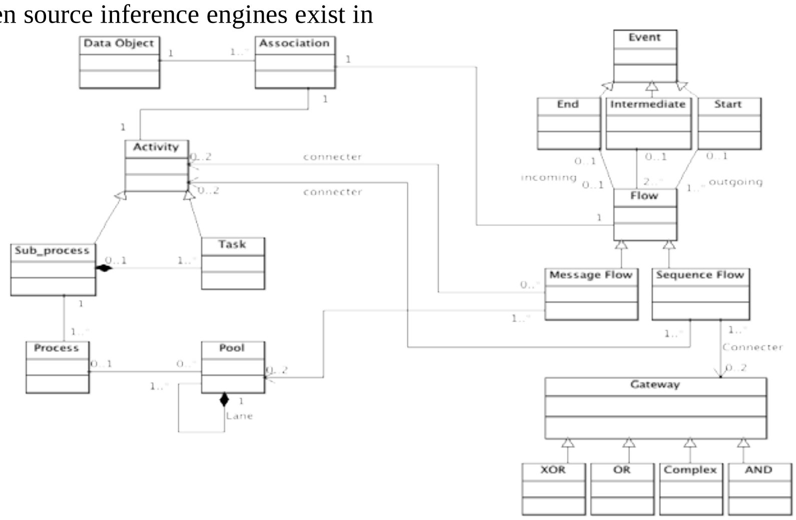

Figure 6: CIM-level BPMN metamodel

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. THE TRANSFORMATION OF THE COLLABORATIVE PROCESS TO SERVICE MODEL

The aim of this part is to transform the CIM Model represented by the BPMN collaborative process to a service oriented process that represents the PIM model. We begin by proposing a PIM meta-model then we suggest a set of mapping from BPMN model to SOA model.

6.1 The SOA MetaModel

The correspondence between these two specifications is difficult to achieve due to the complexity of business processes. The current challenge of BPM is how to provide tools to automate the transition from a business process designed at business level to a process executable at technical level. BPEL (Business Process Execution Language) has been proposed to respond to this need. In fact BPEL is an XML representation of activities related to the process execution. The BPMN describes the process in a static manner while BPEL describes dynamically the process. The code generated from BPMN is unreadable and it is difficult for human to understand and to read it.

These two models are dissimilar and they usually change. On the one hand, the business is growing day after day and it has to respond to the new needs, on the other hand the technology integrates the latest technological developments. This situation usually causes difficulties to maintain perfect synchronization between these two models. So the transition from BPMN model to BPEL model causes several inconsistencies. So we propose a set of mapping rules to transform BPMN process to services

The set of concepts that appear in the PIM-level architecture metamodel are depicted in Figure 7.

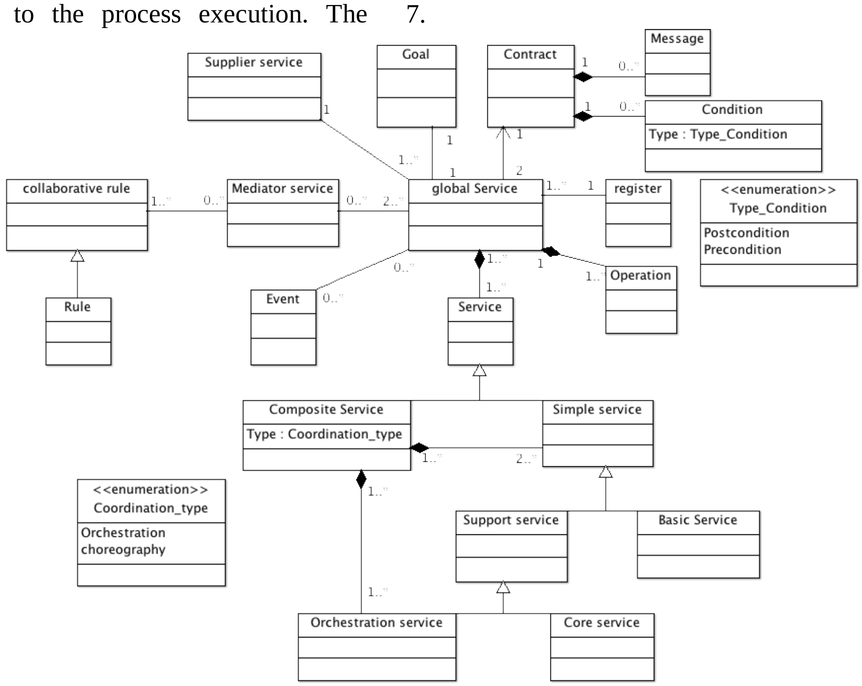

Figure 7: PIM-level SOA metamodel

Services inside a SOA-based system play different roles. They can be classified according to their atomicity in simple services or composite services.

Simple services can be divided into basic services or supporting services. Basic services represent those services encapsulated in a high-level business concept. Its operations are directly related to software functions such as the services providing functionalities for determining the amount of an order. Supporting services perform actions clearly identified and not necessarily with direct relation with the modeled system functionality. Supporting services can be again divided into orchestration services or core services. The orchestration service is a service responsible for coordination and arrangement of multiple services to expose them as a single aggregate service. Core services perform tasks needed for other services to work correctly. For example,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

location services, registry services, security management services, etc.

The mediation service ensures the collaboration of heterogeneous systems services. It is responsible for the composition of services oriented process in accordance with the collaboration rules.

The collaboration rules are employed to drive the development process of collaboration. They describe collaboration constraints and they deal with consistency between heterogeneous services.

Despite the type of service modeled, all of them have a set of operations. The services operations are atomic functionalities that collaborate to outline the service features.

Services communicate and interact through contracts. The service contract is expressed in pre and post conditions that the services involved must respect to work.

6.2 The transformation rules

We defined the deduction rules based on our expertise in BPMN and information system domains and based on some references found in the literature.

The rules are defined by a direct mapping between meta-models elements. The following presents a formal definition of the representation of these rules. We consider the correlation function $x \xrightarrow{gen} y$, where x is a subset of the BPMN process meta-model and y is a subset of the SOA meta-model. This function must be interpreted as follow: ''for every x, detected in the source model, y elements are generated in the target model'' [10].

Rule1: each pool or lane in the BPMN process model is associated to a Supplier service.

$$\forall x \in \{P \cup L\}, x \xrightarrow{gen} y \,/\, y \in SS$$

P: set of pools in the BPMN process

L: set of lanes in the BPMN process

SS: set of service supplier in the SOA metamodel

In the same manner we define the other transformation rules. The mapping between meta-model elements are defined in the table belong TABLE I

Table1: mapping from business process (BPMN) concepts to SOA concepts

Business process (BPMN)	Service Concepts
Pool/Lane	Service Supplier
Process	Global Service
Sub process	Composite service
Activity	Simple Service
Task	Operation
Event	Event
Message	Message flow/ Sequence flow
Gateways	Condition

7. CONCLUSION

The goal of this paper is to present an approach for the interoperability of enterprise systems through the collaboration of their business processes. The approach is based on MDA principles. The collaboration of business processes is achieved at PIM level. To get a collaborative process able to achieve a purpose, we propose a knowledge base system that generates a collaborative process model specific to a given collaboration case. The KBS is designed in accordance with two ontologies and deduction rules that deal with the morphing from collaborative knowledge to collaborative process. Then, the model is transformed to a BPMN process. At PSM level the BPMN process is transformed to a set of services. This transition of models is done according to transformation rules. The suggested approach still has various limitations that have to be improved and several future works are planned. So, on a practical side the approach should be implemented using the technologies presented in each section. User interfaces have to be developed. To validate the approach it should be applied to several use cases. On a theoretical side, various evolutions have to be discussed. The concepts of ontologies should be enriched and other deduction rules have to be proposed to ensure a high level of collaboration success.

REFRENCES:

[1] V. Botta-Genoulaza, P.A. Milletb, A classification for better use of ERP systems, Computers in Industry, 2005,Vol. 56, n. 6, pp. 573-587.

[2] M. Younus, L. Hu, Y. Yong, F. Yuqing, Realization of Manufacturing Execution System for a Batched Process Manufacturing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Industry, Proceedings of the International MultiConference of Engineers and Computer Scientists (Page: 1337-1341 Year of Publication: 2009 ISBN: 978-988-17012-7-5).
[3] X.G Minga, J.Q. Yana, X.H. Wanga, S.N. Lia, W.F. Lub, Q.J. Pengc, Y.S. Mad, Collaborative process planning and manufacturing in product lifecycle management, Computers in Industry, Vol. 59, n. 2-3, 2008, pp. 154-166.
[4] J. Brown, Z. Jiangang, Extended and virtual enterprises—similarities and differences, International Journal of Agile Manufacturing Systems, Vol. 1, n. 1, 1999, pp. 30-36.
[5] R. Jardim-Goncalves, A. Grilo, A. Steiger-Garcao, Challenging the interoperability between computers in industry with MDA and SOA, Computers in Industry, Vol. 57, n. 8-9, 2006, pp. 679-689.
[6] F. J. Moomena, Modélisation des architectures logicielles dynamiques : Application à la gestion de la qualité des services des applications à base des services web, Ph.D. Thesis, National Polytechnic Institute of Toulouse-France, April, 2007.
[7] Object Management Group (OMG), 2006. http://www.omg.org.
[8] Model Driven Architecture, MDA Guide Version 1.0.1, June 2003. http:// www.omg.org/mda.
[9] J. Poole, Model-Driven Architecture: Vision, Standards and Emerging Technologies, 2001.
[10] J. Touzi, Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises, Ph. D. Thesis, Dept. Centre de Génie Industriel - Ecole des Mines d'Albi Carmaux, 2007.
[11] EIF, 2004. European Interoperability Framework, White Paper, Brussels, http://www.comptia.org.
[12] Athena Consortium, 2004. Public document: ATHENA General description v10, http://www.athena-ip.org/.
[13] IDEAS, 2003. A Gap Analysis–Required activities in Research, Technology and Standardisation to Close the RTS Gap—Roadmaps and Recommendations on RTS Activites, IDEAS Deliverables.
[14] e-Gov, 2005. e-Government Unit, e-Government Interoperability frame- work, Version 6.1, 2005.
[15] S. Baïna, Interopérabilité dirigée par les modèles : Une Approche Orientée Produit pour l’interopérabilité des systèmes d’entreprise, Ph. D. Thesis, Dept. Centre de Recherche en Automatique, Henri Poincaré Nancy I University, 2006.
[16] G. Wiederhold, Mediator in the architecture of future information systems, Computer, Vol. 25, n. 3, 1992, pp, 38-49.
[17] S. Chaari, Interconnexion des processus interentreprises: une approche orientée services, Ph. D. Thesis, Dept. National Institut of Applied Sciences, Lyon, France, 2008.
[18] J. Touzi, F. Benaben, H. Pingaud, Model Transformation of Collaborative Business Process into Mediation Information System, Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea (Page: 13857-13862 Year of Publication: 2008 ISBN: 978-3-902661-00-5).
[19] V. Rajsiri, Knowledge-based system for collaborative process specification, Ph.D. Thesis, Dept, Industrial Engineering, Toulouse University, 2009.
[20] A.ESPER, Intégration des approaches SOA et orienté objet pour modéliser une orchestration cohérente de services, Ph.D Thesis, Dept, National Institute of Applied Sciences of Lyon, 2010.
[21] W3C, World Wide Web Consortium. http://www.w3c.org.
[22] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris and D. Orchard, “Web Services Architecture,” W3C Web Services Architecture Working Group note, 2004. http://www.w3.org/TR/ws-arch/
[23] Object Management Group (OMG), 2006. http://www.omg.org.
[24] V. Rajsiri, J-P. Lorre, F. Benaben, H. Pingaud, Knowledge-based system for collaborative process specification, Computers in Industry, Vol. 61, n. 2, 2010, pp. 161-175.
[25] S.Pollard, Collaboration-The Cure-All in New Economy Competitiveness? AMR Research Report, 2002.
[26] T.R.Gruber, A translation approach to portable ontologies, Knowledge Acquisition, Vol.5, n. 2, 1993, pp. 199–220.
[27] S. W. Knoll, T. Plumbaum, J. L. Hoffmann, E. W. D. Luca, Collaboration Ontology: Applying Collaboration Knowledge to a Generic Group Support System, Proceedings of the Group Decision and Negotiation Meeting (Page: 37, Year of publication: 2010, ISBN: 9789056382360).
[28] H. Liu, J. P. Bourey, Transformation from a Collaborative Process to Multiple Interoperability Processes, Enterprise

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Interoperability IV, Enterprise Interoperability IV, 2010, pp.135-144.
[29] Y. Lemrabet, Proposition d'une méhode de spécification d'une architecture orientée services dirigée par le méier dans le cadre d'une collaboration inter-organisationnelle, PH. D. Thesis, Dept. Lille Nord-de-France University, 2012.
[30] T.W. Malone, K.Crowston, G.A. Herman, Organizing Business knowledge – The MIT process Handbook, 2003
[31] R. Benmoussa, R. Derrouich, A. Bouras, Méthode pratique pour le déploiement de l'approche processus au sein d'une chaine logistique, Revue Française de gestion industrielle, Vol. 28, n. 3, 2009, pp. 29-53.
[32] X. Fournier-Morel, P. Grojean, G. Plouin and C. Rognon, Processus métiers et systèmes d'information- Gouvernance, management, modélisation Dunod, Paris, 2011.
[33] Protége´: http://protege.stanford.edu/.
[34] SWRL Specification: http://www.w3.org/Submission/SWRL/.
[35] S.A. Petersen, The Role of Enterprise Modelling in Virtual Enterprises, Collaborative Networks and Their Breeding Environments, Vol. 186, 2005, pp. 109-116.