Model-driven engineering of mediation information system for enterprise interoperability

Frederick Benaben, Sébastien Truptil, Wenxin Mu, Hervé Pingaud, Jihed Touzi, Vatcharaphun Rajsiri, Jean-Pierre Lorre

▶ To cite this version:

Frederick Benaben, Sébastien Truptil, Wenxin Mu, Hervé Pingaud, Jihed Touzi, et al.. Model-driven engineering of mediation information system for enterprise interoperability. International Journal of Computer Integrated Manufacturing, 2018, 31 (1), pp.27-48. 10.1080/0951192X.2017.1379093 . hal-01685412

HAL Id: hal-01685412
https://hal.science/hal-01685412

Submitted on 26 Nov 2018

Model-driven engineering of mediation information system for enterprise interoperability

Frederick Benaben [a], Sebastien Truptil[a], Wenxin Mu[b], Herve Pingaud[c], Jihed Touzi[d], Vatcharaphun Rajsiri[a] and Jean-Pierre Lorre[e]

[a]Centre Génie Industriel, Mines Albi-Carmaux, Université de Toulouse, Campus Jarlard, Albi, France; [b]Department of Information Management, School of Economics and Management, Beijing Jiaotong University, Beijing, China; [c]Centre Universitaire Jean-François Champollion, Albi, France; [d]JPL Informatique SA, Saint Sulpice, Suisse; [e]Linagora, Toulouse, France

ABSTRACT

The MISE approach (mediation information system engineering) aims at defining and designing a platform, dedicated to initiate and support any collaborative situation among potential partners. Collaborative situations may emerge from manufacturing contexts (industrial production partners), economic contexts (supply chain), social contexts (crisis management partners) and any other contexts in which a set of organisations should work altogether to reach common and individual goals, exploiting each others competencies. The MISE approach is based on a model-driven engineering vision (MDE) dedicated to designing a mediation information system (MIS) in charge of improving interoperability in emerging collaborative situations. This MIS is dedicated to supporting the collaborative behaviour of the collaborative network by dealing with exchanged data, shared services and collaborative workflows. The final objective is a mediator system able to manage the operational collaboration of partners, through there information system, without constraint (at least with as less constraint as possible). The MIS design crosses the different abstraction layers of design (business, logical and technological) and exploits the associated models at each level to build the models of the next level. This paper presents the models involved (dedicated to the computer independent model, platform independent model and platform specific model levels of the MDE approach) and the transition mechanisms between levels.

1. Introduction

The ability of organisations (enterprises, institutions or others) to operate in collaboration is a key indicator of their level of competitiveness. However, organisations, whether recent or well established, rarely take much interest in their collaborative capacities (which are mainly empirical abilities, informally built and used as needed) (Agostinho et al. 2016). In this paper, the specificities and requirements of organisational collaboration from an information system (IS) point of view are studied, in order to define an engineering approach dedicated to facilitating the deployment of these emerging collaborative networks. Section 2 presents an overview of the research challenges addressed in this paper (including a literature review). Section 3 is dedicated to describing the overall approach in general terms. Section 4 presents some preliminary elements that should be taken into account. Section 5 describes the model-driven engineering (MDE) approach in detail. The final section concludes and draws some perspectives.

2. Positioning of MISE approach

This section is structured according to three steps: (1) the general context of these research works, dedicated to identify some precise considerations that should be taken into account to refine the description of the main objective of these research works; (2) the specific refinement of the main objective of mediation information system engineering (MISE) approach according to the previous considerations and (3) a literature review regarding some useful scientific or technological components, covering the various facets of the presented approach.

2.1. Considerations about the context

The general object of this paper concerns *defining, formalising* and *equipping collaborative situations*. In this field of study, four main developments are noticeable:

- Organisations used to find their partners in a restricted geographical area (for practical reasons). Nowadays, improvements in logistics, the use of Internet, the use of the English language and the global market have removed this limitation.
- Organisations used to build their collaborative links slowly and carefully (for cultural reasons). Nowadays, increased economic competition and the need for reactivity have forced a change in this approach.
- Organisations used to build stable and long-term collaborations (due to a lack of requests). Nowadays, the

need to seize/create sporadic opportunities has led to a change in the environment.

- People (managers, bosses) used to be the interface of organisations (exchanging with other organisations) as well as the functional backbone of organisations (driving process and business activities). Nowadays, ISs and computed systems have taken over these functions (at least partially).

From the previous four statements, it is easy to infer that the main idea behind the main goal of these research works (*defining, formalising* and *equipping collaborative situations*) is the management of interoperability of organisations. Actually, the previous statements mainly define new expectations for collaborations: farer, faster, more opportunistic and more computerised collaboration ability. These are the features for integration through IS interoperability. *Interoperability* is defined in Konstantas et al. (2005) by the European network of excellence InterOp as 'the ability of a system or a product to work with other systems or products without special effort from the customer or user'. It is also defined in Pingaud (2009) as 'the ability of systems, natively independent, to interact in order to build harmonious and intentional collaborative behaviours without modifying deeply their individual structure or behaviour'.

In addition, this paper claims that a network of organisations can be seen as a system of systems (SoS). Actually, according to Maier (1998), there are five criteria to define a SoS:

(1) Operational independence of the elements (systems)
(2) Managerial independence of the elements (systems)
(3) Evolutionary development of the SoS
(4) Emergent behaviour of the SoS
(5) Geographical distribution of elements.

The collaboration of organisations concerns elements (organisations), which are, by definition, independent on both the *operational* level (different skills) and on the *managerial* level (different heads). Such an emerging collaborative situation is an *evolving entity*. New partners may join the network while some other may quit it. Moreover, the *behaviour* of the collaboration should *follow* the evolutions of the collaborative situation. Finally, the collaborative network may be widely *distributed* insofar as some partners could be in completely different places. Consequently, collaboration of organisations meets these five criteria and may be considered as a SoS.

Finally, there are (at least) two main conceptual architectures able to support interoperability (Wiederhold and Genesereth 1997): pear-to-pear structure (where each partner is expected to communicate efficiently with the others) and mediation structure (where a third-part component is in charge of partners' interoperability). On the one hand, pear-to-pear architecture requires either the partners to be able to connect efficiently with all involved organisations (including data translation, service access, process sharing): common languages and conventions may be required (which is not very good for reactivity of the network), or each partner should be able to 'speak' the language of all the others (which is not very good for dynamicity of the network). On the other hand, mediation architecture, as presented in Wiederhold and Genesereth (1997), requires a mediator, which will perform all the interoperability functions, but which needs definitely to be (1) designed and deployed accordingly with the considered situation (to ensure reactivity) and (2) maintained accordingly with evolutions and forks of the living situation (to ensure dynamicity). In this paper, the mediation architecture has been chosen, which imply to define a mediator (mediation information system [MIS]) able to support on the fly the emerging collaboration.

2.2. Main objective of MISE research work

Regarding the considerations presented in Section 2.1, the general purpose of this paper can be refined (*defining, formalising* and *equipping collaborative situations*) into a more complete and precise goal: *the definition of an efficient engineering approach for agile MISs dedicated to supporting interoperability of ISs inside a collaborative network, seen as a SoS*. In the way the objective is formulated, all the previously enumerated aspects can be found.

Such an objective requires defining an engineering approach, exploiting the adequate collaborative knowledge to design a MIS according to a technical architecture. These three elements (*engineering approach, collaborative knowledge* and *technical architecture*) are the main components that should be studied, selected and defined in order to describe the MISE approach. Consequently, in the next section, the state of the art focuses on these three components.

2.3. State of the art and foundations

The following three next paragraphs are dedicated to provide hints about the three main components of the MISE approach: *engineering approach, collaborative knowledge* and *technical architecture*.

Regarding engineering approach, model-driven architecture (MDA), fostered by the Object Management Group, is a very accurate approach to deal with system complexity through abstraction levels (OMG 2003). MDA defines concepts such as model, metamodel (MM) and transformation principles.

In the research field of manufacturing, MDA is widely used. Obermeier, Braun and Vogel-Heuser (2015) develop the modular automation for reuse in manufacturing systems (modAT4rMS) approach to support the MDE of object-oriented manufacturing automation software with regard to its usability and software modularity. Estevez et al. (2016) explore the advantages that MDE provides for the development of applications for robotic manipulators' platforms. Specifically, a modelling approach is developed to generate the target code automatically. Tchoffa et al. (2016) presents a new way of combining model-based enterprise platform engineering, MDA, and system engineering in order to address the establishment of a sustainable interoperability within Dynamic manufacturing network. Lundgren, Hedlind and Kjellberg (2016) present model-based approach for integrated process planning and quality assurance, which enables new functionalities and provides more efficient support to production engineering processes. Gao et al. (2016)

reduce the problem size of workflow dynamic changes, decrease the computing complexity of workflow regions recognition and have flexibility to take the different migration policies corresponding to the different workflow running instances dynamically.

MDE is based on the foundation of MDA and provides a development process: different levels of models are organised in the modelling space. Kent (2002) provides a clear description of the MDE paradigm and its structure: *computer independent model* (CIM) describes the target system from a business point of view. *Platform independent model* (PIM) describes the same system from a software point of view but only according to functional considerations. Finally, *platform specific model* (PSM) describes one possible implementation of the previously obtained PIM (including considerations about languages, computing platforms, integration middleware, communication networks etc.). According to these considerations, MDE may be used as a backbone for MIS design (i.e. engineering approach).

Concerning the *collaborative knowledge* that should be used all along the MDE engineering approach, the real question is 'what model(s) should be used/built along the engineering approach?'. To answer that question, it is necessary to state that (1) the MISE approach aims at gathering knowledge regarding a target enterprise network and (2) the MISE approach aims at designing a collaborative IS. Based on these statements, both 'enterprise modelling' and 'IS modelling' should be considered.

Vernadat (1996) presents enterprise modelling according to four points of view: *informational* view (data and knowledge of enterprises), *resource* view (capabilities, materials and human resources), *organisational* view (roles, responsibilities, sectors of enterprises) and *functional* view (events, flows, processes and activities). The following picture (Figure 1) presents the enterprise MM extracted from Vernadat (1996):

Activity is the central concept of this MM (not only because it is in the middle of the diagram, but mainly because this concept is linked to a lot of others and especially concepts of all other packages).

Regarding IS modelling, Booch, Jacobson and Rumbaugh (2004) suggest that the classical splits *static view/dynamical view* and *logical view/physical view* could be refine according to four views: *functional* view (different from the functional view of enterprise modelling, describing functions and abilities of the system), *structural* view (representing the static and logical views of the system, i.e. layout of logical components), *behavioural* view (dynamic and physical views of the system, describing exchanges and operating modes of components according to scenarios) and *architectural* view (static and physical views of the system, presenting the technical organisations of components).

Considering, first, that processes (from the *functional view* of enterprise modelling) are strongly connected to other components of enterprise modelling (thanks to the concept of *activity*) and, second, that the *behavioural view* of IS modelling is also closely coupled with the other views (because it shows the global behaviour of components from these other points of view). The first conclusion of this section is the following: *Process modelling* might be a very relevant pivot to bind *enterprise modelling* with *IS modelling*. This statement is perfectly in line with considerations of Anaya et al. (2010) about Unified Enterprise Modelling Language (UEML): The central motivation of UEML is to 'exchange information contained in enterprise and IS models across modelling language boundaries'.

As for the IS architecture, and considering the previous statement, a good candidate might be the one strongly based on process modelling. Schematically, service-oriented architecture (SOA) is a paradigm that describes an IS architecture according to a set of services (applications) that can be invoked through a middleware (in a

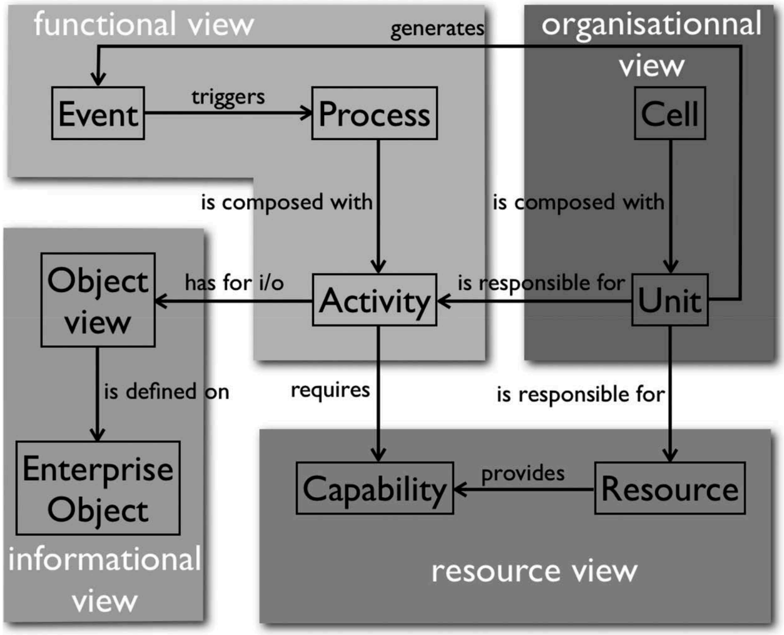

Figure 1. Enterprise metamodel (from Vernadat 1996).

loose binding philosophy). Services may be added or removed easily. Furthermore, in some SOA contexts, workflow engines allow to execute workflows (implementation of processes). This makes SOA one of the most 'process-centred' IS architecture. The PIM4SOA project (Benguria et al. 2006) defines four points of view for SOA paradigm: *service view, process view, information view* and *quality of service view*.

The synthesis of the previous considerations is the following: gathering knowledge about the collaborative situation should be based on enterprise modelling principle in order to build collaborative process model. This process model could be transformed (in a model-driven approach) into an SOA IS model.

3. MIS engineering: theoretical overview

3.1. Main hypothesis

The main hypothesis of this paper concerns the supposed service orientation of organisations' ISs. The SOA philosophy is one main pillar of the work presented here. According to Morley, Hugues and Leblanc (2002), the computed part of the IS embeds the concepts of *processes, functions* and *information* inside a technological architecture, which runs *workflows*, which use *services*, which need *data*. In other words, in order to be potentially integrated into the collaboration, each organisation must be able to expose (and share) its capabilities (so that they can be used by the collaborative network to perform specific feature useful to reach the collaborative goals). On a technological point of view, this means that partners should be able to expose (1) their applications as web services (compliant with usual protocols such as SOAP or REST) and their human-based abilities as (2) web-service interfaces (so that people can be invoked appropriately to perform their duty).

As a consequence, the final contribution of the research work presented in this paper can be figured as 'an approach to deploy a mediation system able to (1) formally define and (2) operationally orchestrate the collaborative behaviour schema involving partners that are SOA compliant'.

3.2. Interoperability in SOA context

According to Aubert and Dussart (2002) and Bénaben et al. (2006), in such a context of IS structure, interoperability may be declined according to three main functions:

- Transfer and translation of *data*
- Management of *services*
- Orchestration of collaborative *workflows*.

Because it seems that partners' ISs cannot naturally assume these functions (except with strong constraints of technical standardisation), the proposed approach, using mediation, aims at providing a third system in charge of these functions: the MIS (Figure 2). This MIS should meet with two requirements:

- *It should be technically able to connect to the services and data of each partner organisation*. The SOA context allows partners to expose the subset of data, services and workflows (seen as services) they agree to share. The MIS has therefore at its disposal a certain amount of knowledge (syntactic and potentially semantic) concerning the data of the collaboration and a repository of the shared services, usable in the collaboration.

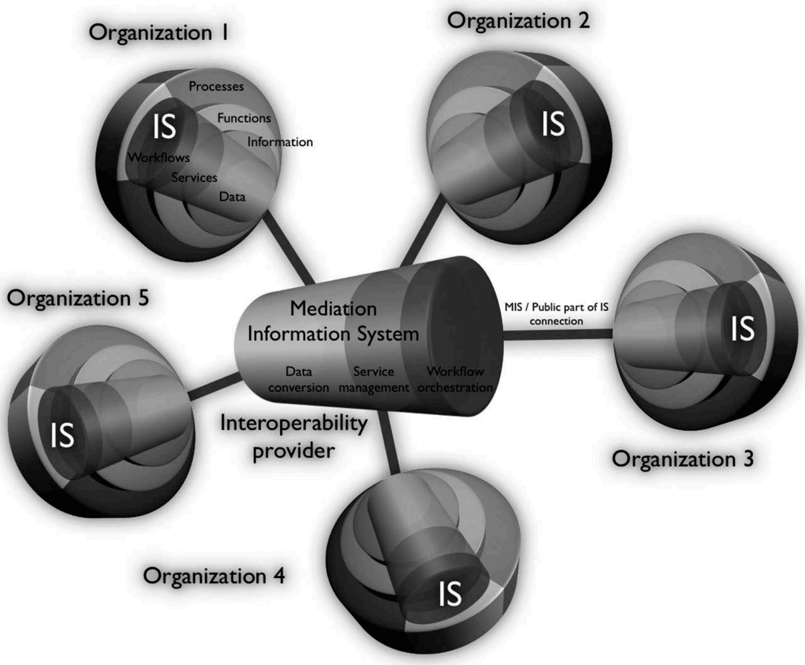

Figure 2. ISs interoperability through mediation IS.

- *It should be technically able to orchestrate the collaborative behaviour as a workflow*. The MIS should include an executable model of the collaborative workflow and an orchestration engine capable of running that behaviour.

This conceptual view of the MIS introduces the several levels of abstraction that the engineering approach must cross:

- *Business*: level where the collaboration is identified and defined in functional and conceptual terms.
- *Logical*: level where the MIS structure and its behaviour are defined.
- *Technological*: level where the final mediation system is deployed according to the previous requirements.

3.3. Big picture of the MIS engineering methodology

This dive into abstraction layers is based on MDE and on the associated model transformation concepts (OMG 2003). The general principle is therefore to gather a structured knowledge on the considered collaboration (including the involved organisations, their respective roles, the topology of the network, the collective or individual goals, the shared services and data etc.) in order to instantiate a specific ontology. By applying deduction rules on this knowledge, a collaborative process model is deduced (as described in Rajsiri et al. 2010): The main principle is to use the knowledge base to find appropriate partners' functions to reach each identified objective. This is the CIM level, corresponding to the business layer. The knowledge embedded in this collaborative process is then used through model transformation mechanisms in order to propose a MIS structure (as described in Touzi et al. 2009) at the PIM level, corresponding to the logical layer: The main principle is to map components of the business process model (activities, lanes, pools, events) onto concepts of the SOA architecture (services, partners, data). Finally, a technical model of the MIS is build, based on the knowledge embedded in the PIM and on the technological target platform, which is an ESB (enterprise service bus) at the PSM level, corresponding to the technological layer. The following picture (Figure 3) illustrates this global approach of MDE:

Before detailing this MDE approach, the next section will focus on the required theoretical elements: model transformation principles, MMs and technology.

4. Conceptual requirements to the MDE approach

The main idea is that the approach for *MISE* (which is at the centre of these research works) is based on model-driven principles. Therefore, this approach aims at using *model transformations* to dive into the following abstraction levels: business, logical and technological. More precisely, the goal is to move from business level to logical level, and then from logical level to technological level. However, to move from a source level to a target level, it is required to formalise the knowledge at the source level (by instantiating the concepts of the source level *MM*), then to apply model transformations (onto the obtained model of the level) to obtain a new model (i.e. instances of the concepts of the target level *MM*). Consequently, as far as Section 5 is dedicated to describe this path across the abstraction levels by presenting the transformation rules between each of the levels, the current section presents all the preliminary theoretical prerequisites that should be assimilated to describe efficiently the whole approach in Section 5. This section

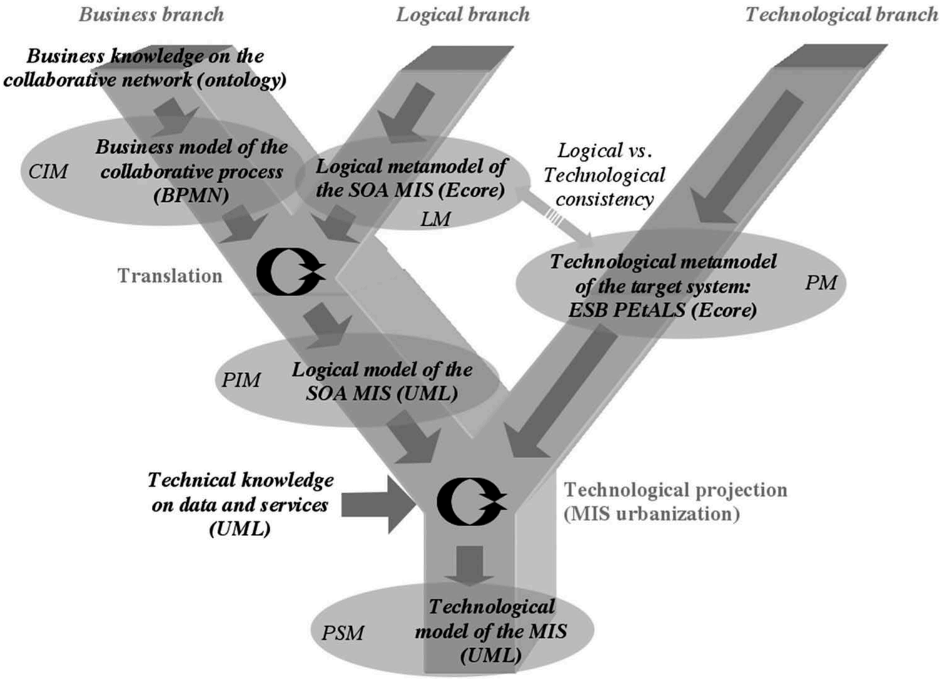
Business branch
Logical branch
Technological branch
Business knowledge on the collaborative network (ontology)
CIM
Business model of the collaborative process (BPMN)
Logical metamodel of the SOA MIS (Ecore)
LM
Logical vs. Technological consistency
Translation
Technological metamodel of the target system: ESB PEtALS (Ecore)
PM
PIM
Logical model of the SOA MIS (UML)
Technical knowledge on data and services (UML)
Technological projection (MIS urbanization)
PSM
Technological model of the MIS (UML)

Figure 3. Overview of the model-driven engineering approach.

introduces, first, the general principle of model transformation that is used in MISE, second, the three required MMs that are used at each level during model transformations and, third, the technological platform supporting the whole approach.

4.1. *Model transformation in a model-driven approach*

Considering our model-driven issue, the first crucial point of the present approach is model transformation, which may be synthesised as follows (Figure 4):

A *source model* is used (built according to a *source MM*). The key point is that the *source MM* shares part of its concepts with the *target MM*. Note that the two spaces, source and target, have to be partially overlapping in order to allow model morphism). As a consequence, the *source model* embeds a *shared part* and a *specific part*. The *shared part* provides the *extracted knowledge*, which may be used for the model transformation, while the *specific part* should be saved as *capitalised knowledge* so as not to be lost. Then, *mapping rules* (built according to the overlapping conceptual area of MMs) can be applied onto the *extracted knowledge* in order to provide the *transformed knowledge*. That *transformed knowledge* and an *additional knowledge* (to fill the lack of knowledge concerning the non-shared concepts in the *target MM*) may finally be used to create the *shared part* and the *specific part* of the *target model*. Note that both *capitalised knowledge* and *additional knowledge* may be empty depending on the model transformation case under consideration.

4.2. *Business MM*

A preliminary discussion is required here, concerning the content of this MM. Dedicated to the business level (i.e. the static and dynamic characterisation of the collaborative network), this MM is directly linked to the field of enterprise modelling. A study concerning the covering, by a BPMN process model, of the points of view of enterprise modelling proposed by Vernadat (1996) is made in Bénaben et al. (2006) and explained in detail in Touzi (2007). This study asserts that such a model suitably covers the *functional view* and also partially covers the *informational* and *organisational* views. It does not cover the *resources* view to any great extent. Considering these observations, the study concludes that this covering, even if incomplete, may be considered as sufficient.

The business MM is thus focused on capitalising the business knowledge into a collaborative business process model, involving all the partners of the collaborative network. This MM (presented in Figure 5) is based on BPMN language and organises the collaborative situation according to the mediation principle exposed in Wiederhold and Genesereth (1997). This mediation principle claims that the mediator should be able (1) to communicate with each of the members of the network and (2) manage the partners to reach altogether their collaborative goals (or the best compromise between their individual goals). The MIS (or CIS – for collaborative information system – as it is named in the following figures) orchestrates the whole behaviour by coordinating the services of all partners:

A collaborative process respecting this MM is then structured according to one pool for the MIS (*CIS pool*) and one pool per partner (*partner pool*). The dynamic aspect of the process is embedded in the MIS pool through mediation services (*CIS task*), events (*event*), gateways (*gateway*) and internal flows (*sequence flow*) carrying data (*data*). Partners propose their own services (*partner task*) and communicate exclusively with the MIS through external flows (*message flow*) also carrying data (*data*).

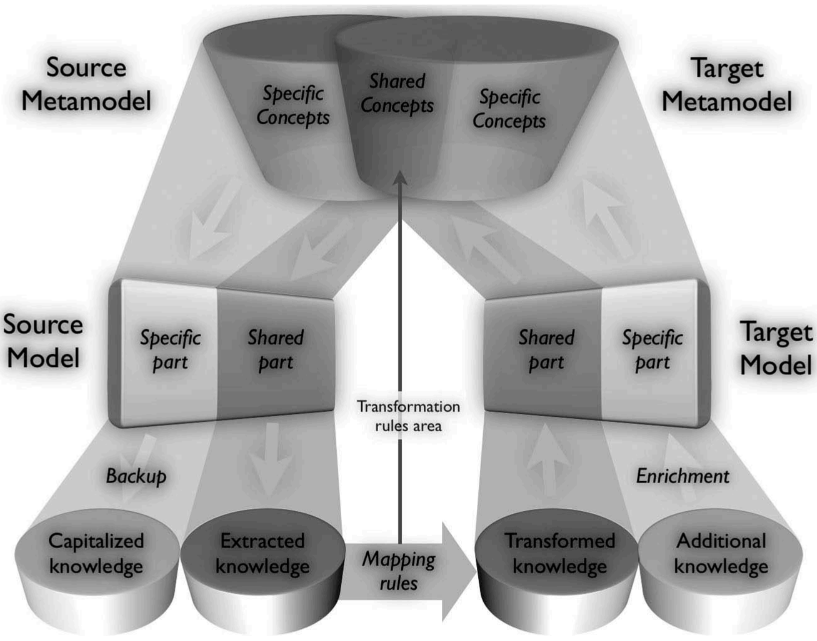

Figure 4. Model transformation principle.

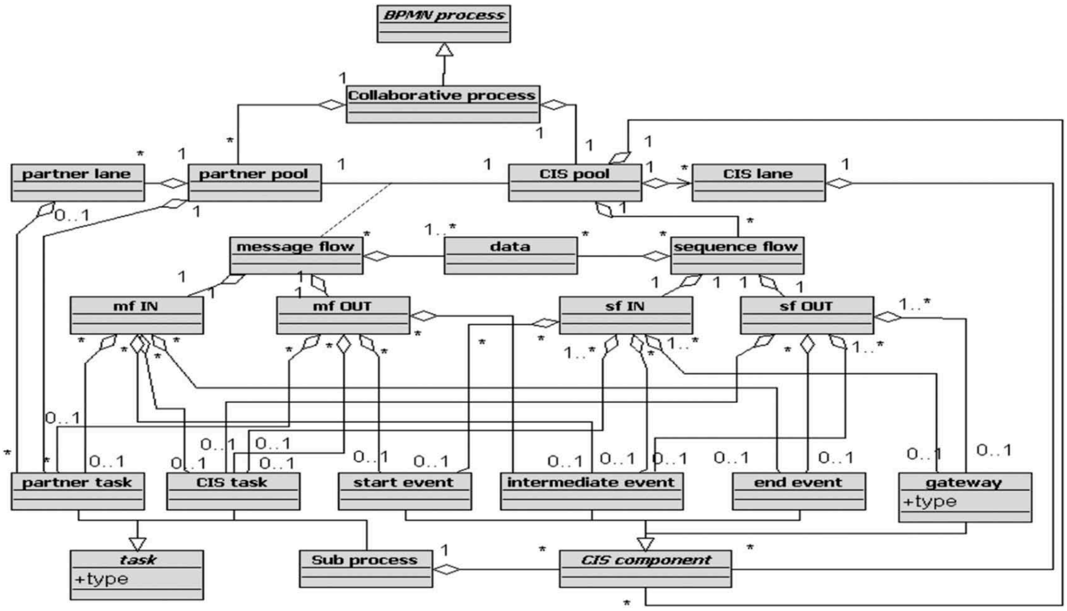

Figure 5. Business metamodel: collaborative process in BPMN.

Finally, the presented business MM is dedicated to be the support for the description of the collaborative behaviour. The objective of the usage of such a MM is that any model based on that MM embeds a knowledge content that can be used to infer the structure of the appropriate mediation IS.

4.3. Logical MM

This second MM concerns the logical architecture of the MIS. It is mainly based on the PIM4SOA model presented in Benguria et al. (2006). The mediator architecture (presented in Figure 6) is organised according to three packages: *services view* (functions), *information view* (business objects) and *process view* (behaviour).

The *services* view contains partners' services (listed in a services repository) and mediation services (CIS services on Figure 6). These services may be divided up into operations. The *information* view contains all the business objects that may be used in the collaborative process (and their semantic/format features). Finally, the *process* view embeds all the elements needed to describe the workflow, in a language close to business process execution language (BPEL, which is the target orchestration language). The choice of BPEL is due to the fact that some technologies for the implementation of modern IS (for instance those based on *Java Business Integration* [JBI]) include BPEL engines.

Finally, the presented logical MM is dedicated to be the support for the description of the structure of the MIS. The objective of the usage of such a MM is that it provides the structural knowledge required to describe an efficient SOA IS structure, able to support the collaborative behaviour previously described using the business MM.

4.4. Technical MM

The technical MM (presented in Figure 7) is also based on SOA aspects but it contains more information than the logical MM. Although both the MMs have the same overall structure, they are different in some details.

The main differences between this technical MM and the previous logical MM are listed below:

- The class *partner service description* includes seven new attributes dedicated to embedding the technical knowledge contained in Web Service description Language (WSDL) files.
- The class *service* includes one new attribute dedicated to assuming the matching between business service and technical service.
- The class *operation* includes two new attributes, *input* and *output*, dedicated to representing the incoming and outcoming business objects of this operation.
- The class *business object* includes one new attribute dedicated to assuming the matching between general business message and specific web service message.
- The class *format* includes two new attributes dedicated to representing the name and type of the element.
- The class *collaborative business process* includes six new attributes dedicated to embedding the technical knowledge for the BPEL file.

The associations *in* and *out* between *service* and *business object* classes are now between *operation* and *business object* because one service may have several operations and *business object* should be linked to *operation*.

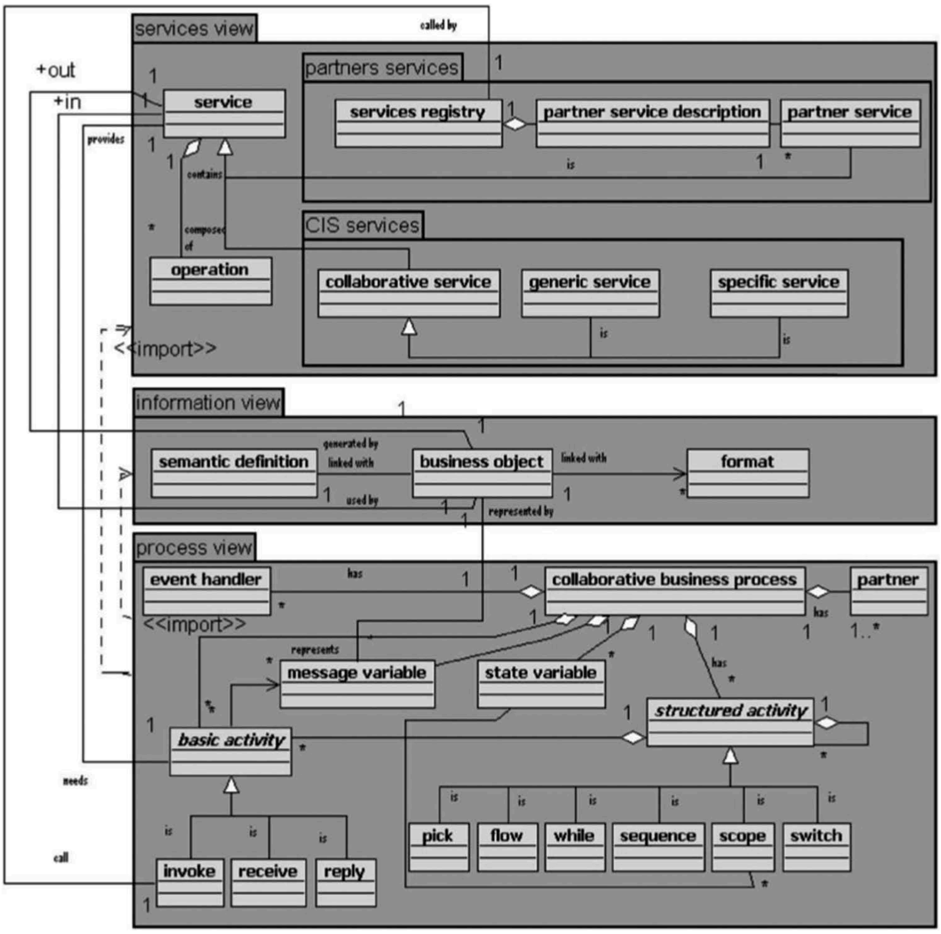

Figure 6. Logical metamodel: SOA architecture of the MIS in UML.

Finally, the presented technical MM is dedicated to be the support for the description of the deployment infrastructure of the MIS based on PEtALS ESB. The objective of the usage of such a MM is that it provides the technological knowledge required to describe an efficient MIS IS infrastructure.

4.5. Target technological platform

The definition of a target technological platform is not critical in the whole approach. Actually, it has been decided to introduce the one that has been used for these research works to make it more concrete and to show feasibility and applicability. The most important thing on the technological dimension is to use an SOA platform. Furthermore, one important aspect is that this platform should, on the one hand, support the MIS engineering tools (model transformation tools mainly) and, on the other hand, should be able to connect all web services of partners. Consequently, it is important for the platform to be compliant with the 'design-time' tools, but it is mandatory for that platform to be as much open as possible to support all kind of partners' services.

In the context of these research works, the target technological platform is the open-source ESB PEtALS (from OW2 and EBM WebSourcing), which is based on the JBI standard from JSR 208. This standard is specifically dedicated to defining the service containers and the way they behave. Several existing ESBs, open-source or proprietary, are based on JBI (*Open ESB* from Sun, *Servicemix* from Apache) or compliant with the JBI standard (*Sonic ESB* from Sonic Software or *Mule* from Mule Source). All these platforms could be used to run the MISE approach. The overall principle of this software is presented in the following picture (Figure 8):

PEtALS ESB aims at linking heterogeneous services, which are either 'fully compliant with the bus' and then embedded in the bus (as service engines [SEs]), or compliant with some specific communication protocols (SOAP, http etc.) and then located outside the bus. All these services will be invoked by the workflow (according to the behavioural model) through the bus, thanks to the SE dedicated to orchestrating the workflow.

Each operational service, whether internal or external, is represented by a *service unit* (SU) in charge of defining its specific interface (especially its inputs and outputs, by means of a *WSDL* file, for instance). Each SU is embedded in a *service assembly* (SA), which provides the link to the operational service. If the service is a SE, i.e. embedded in the bus, then this link is a direct link. If the service is an external *service*, then this link is an indirect link, through a *binding component* (BC) in charge of providing the messages within the appropriate format. Thus, if several services use the same communication protocol, their SU might be included in the same SA, pointing to the same BC.

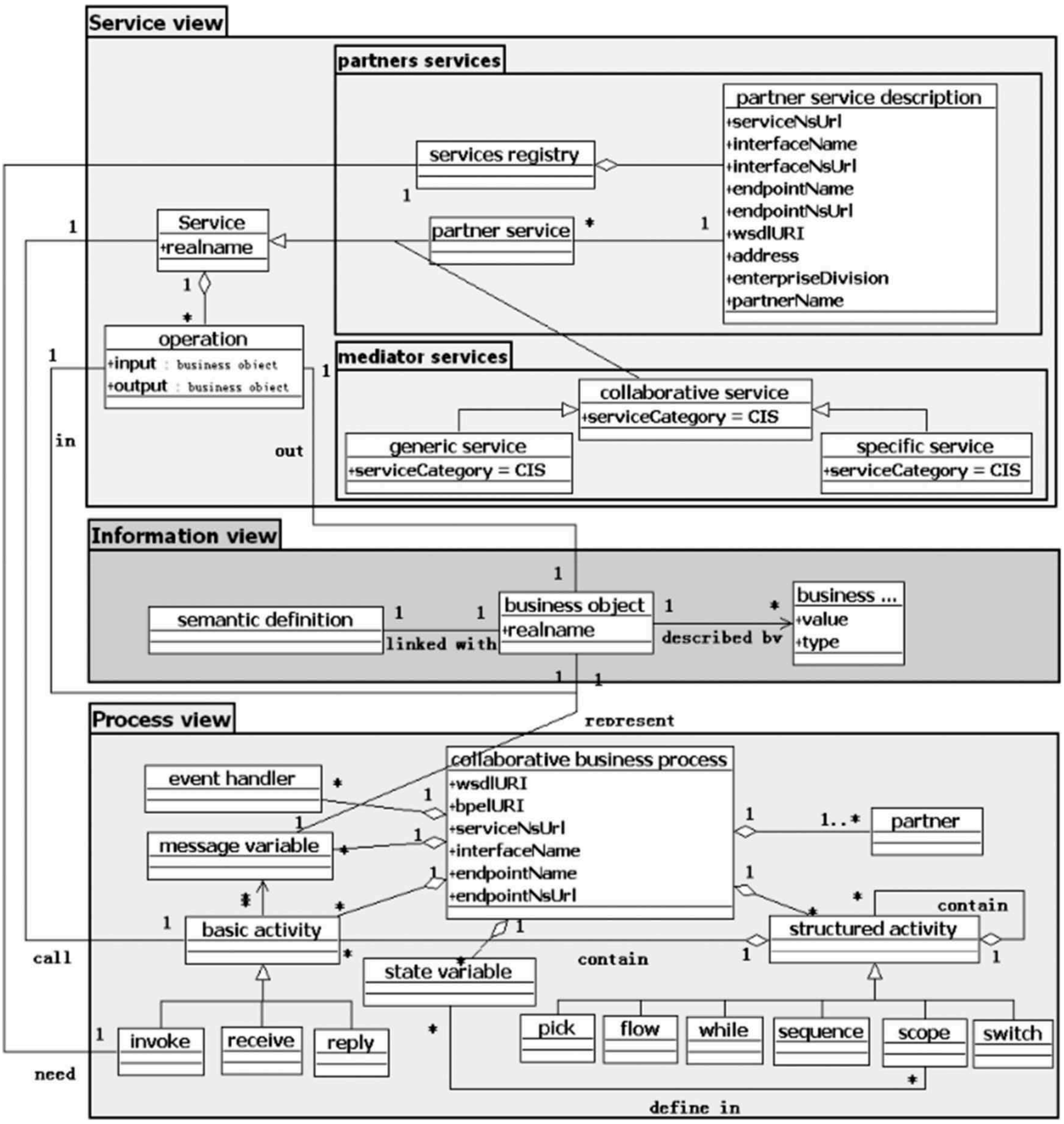

Figure 7. Technical metamodel: a more complete SOA architecture of the MIS in UML.

There is a specific SU (let us call it the SU-BPEL), which contains the workflow model (in BPEL format). This SU is included in a specific SA (let us call it the SA-BPEL) connected to the orchestration SE (let us call it the SE-WF). The SU-BPEL, through the SA-BPEL, provides the SE-WF with the workflow model. The SE-WF contacts the SUs (through their SAs) in order to invoke the services, according to the workflow model.

Finally, in order to deploy this middleware correctly, four types of elements must be deployed with ESB PEtALS:

- All the required SEs: These elements will be necessary for the deployment of SAs that contain SUs linked to these internal services (including the specific SE-WF dedicated to running the BPEL orchestration).
- All the required BCs: These elements will be necessary for the deployment of SAs that contain SUs linked to external services through protocols managed by these BCs.
- All the SAs: These elements embed all the SUs that define the interface of all the required (internal or external) services. They are connected to one SE or one BC. In order to facilitate the evolution of the set of supported services, several SAs might be potentially connected to the same SE or BC (easier to replace).
- The specific SA-BPEL: This element, connected to the SE-WF, contains the SU-BPEL, which embeds the workflow models and also a specific file (named ProcessArtifact) that is in charge of providing the SE-WF with the physical address of each external service.

Based on these files, an *Ant* file is defined (able to start the installation of all the previously listed files). By means of this deployment file, an instance of PEtALS may be deployed, specifically dedicated to the collaborative situation.

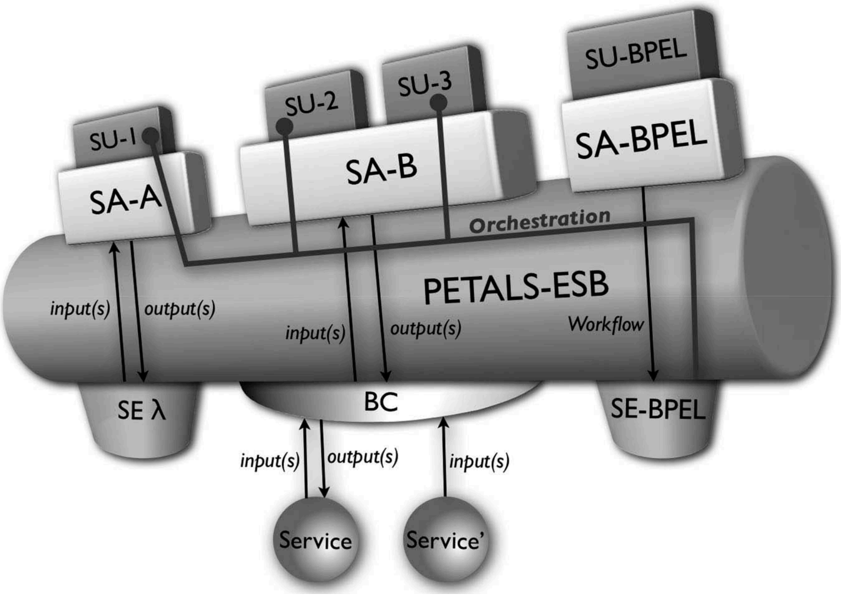

Figure 8. General principle of PETALS ESB.

5. MDE approach

First, this section describes the three abstraction levels of the MDE approach (CIM, PIM and PSM) and the concrete model transformations of each level (based on theoretical elements described in Section 4.1 and specific MMs presented in Sections 4.2–4.4). Then, the deployment mechanisms and agility features are described on the basis of the technological architecture introduced in Section 4.5. Finally, the question of agility is treated in the last part.

5.1. CIM level

The general principle of this first step (described in Rajsiri et al. 2010 and detailed in Rajsiri 2009) is the following: *using an ontological approach to deduce, from the characterisation of a collaborative situation, the business model of a collaborative process (respecting the MM described in Section 4.2)*.

A collaboration of organisations may be characterised through its *common goal*, its *actors*, their *relations*, the *topology(ies)* of the network etc. (Rajsiri 2009). The design of the CIM in BPMN should thus be based on these elements (instantiated, completed and refined). A collaborative network ontology (CNO in OWL-DL), separated into two interconnected parts, has been produced (Figure 9). The first part contains elements describing the collaboration (CO for collaborative ontology), while the second part is dedicated to the concepts of the collaborative process MM (CPO for collaborative process ontology).

The higher part (CO) is instantiated by gathering knowledge from the studied situation (goal, actors, relations, topology etc.). The lower part (CPO) is initially populated with instances extracted from the MIT Process HandBook (Malone, Crowston, and Herman

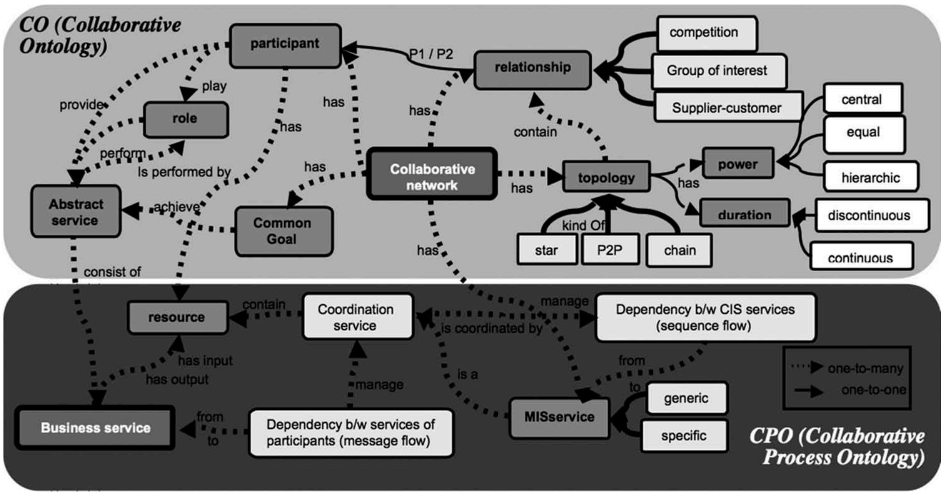

Figure 9. Graphical representation of the collaborative network ontology (concept level).

2003) (which is a collection of more than 7000 activities, sub-activities and exchanged information involved in business processes). Even if this knowledge is abstract and disconnected from any specific case, it is usable to define a generic view of the behaviour (which should later be linked with the real services of the actors). By applying some SWRL deduction rules, described in Rajsiri (2009), instances of the CPO may be deduced from instances of the CO. These rules may be classified according to four groups:

- The first group of rules is intended to derive an abstract service and role when each is provided. According to Petersen (2005), each organisation is represented by its goals, the activities to achieve the goals, the roles that perform the activities and the skills that are required to fill the roles. This definition refers to the relation between role and activity. This activity can be considered as an abstract service, which describes the skills of its provider. The rule that derives abstract service when role is recognised may be described thus:

Participant(?x) ∧ playRole(?x, ?y) ∧ performAService(?y, ?z) → provideAService(?x, ?z)

This rule starts by retrieving roles of participant and finding abstract services that can be performed by these roles. Then, the rule will return the list of abstract services that correspond to the roles the participant plays.

- The second group of rules concerns the deduction of business services when an abstract service is recognised. This rule is defined on the basis of the BAM of the PH (Malone, Crowston, and Herman 2003), which states that every abstract activity has its corresponding functional-level activities. These functional activities can be considered as business services. There is only one rule in this group:

Participant(?x) ∧ provideAService(?x, ?y) ∧ hasBusinessService(?y, ?a) → provideBusinessService(?x, ?a)

This rule starts by retrieving business services that correspond to the abstract services provided by the participant. Then, the rule will return the list of business services that the participant should expose. This rule is the key that creates the semantic connections between the CO and CPO via the concepts of abstract and business services.

- The third group concerns the deductions of dependencies for both message and process logic sequence flows, coordination services and MIS services. The rules defined in this group are the most complicated ones because they take into account several concepts at the same time. Below is an example of the rules of this group. This rule allows us to deduce dependencies when two business services belonging to different participants have a resource in common. The resource dependency concept shows that coordination services should be inserted to take into consideration the exploitation of resources. Such coordination services can be considered as MIS services.

CNetwork(?a) ∧ hasRelationship(?a, ?z) ∧ P1(?z, ?y) ∧ provideBusinessService(?y, ?c) ∧ hasOutput(?c, ?d) ∧ P2(?z, ?x) ∧ provideBusinessService(?x, ?b) ∧ hasInput(?b, ?d) ∧ CoordinationService(?f) ∧ manipulateResource(?f, ?d) ∧ Dependency_between_BusinessServices (?e) →

fromBusinessService(?e, ?c) ∧ toBusinessService(?e, ?b) ∧ containResource(?e, ?d) ∧ isCoordinatedBy(?e, ?f) ∧ hasMISservice(?a, ?f) ∧ MISservice(?f)

This rule starts by finding a relationship between two participants via P1 and P2 relations. Each participant provides its own business services which have input and output resources. The rule verifies whether the output of a business service is the same as the input of other business services. If so, the rule finds a coordination service that can manipulate such a resource and create dependency between these two business services. It also defines this coordination service as a MIS service.

- The fourth group is dedicated to deducing a list of abstract services to be included in the network. The abstract services deduced by the first rule are the ones that the involved partners provide to the others. They are a subset of the abstract services obtained by this actual rule. There is only one rule in this group which derives abstract services from goal:

CommonGoal(?x) ∧ description(?x, ?a) ∧ swrlb:substringBefore(?y, ?a, " ") ∧ AbstractService(?b) ∧ name(?b, ?c) ∧ swrlb:containsIgnoreCase(?c, ?y) → achievesAService(?x, ?b)

The rule starts by segmenting the description of common goal and keeping only the first word found. Then, the rule searches in the KB for abstract services whose name contains this word. The abstract services obtained are the services that concern all involved partners and which the network itself has to provide.

Instances generated using these rules are extracted through SPARQL requests. These instances may then be injected, through an XSLT style sheet, into a BPMN file. The obtained collaborative process respects the MM described in Section 5.1 and includes the specificities of the considered collaboration.

Regarding Section 4.1, the CIM model is obtained using a model transformation mechanism where the *source MM* is the characterisation of the collaborative situation included in the CO, the *target MM* is the collaborative process MM while the *additional knowledge* is the knowledge extracted from the MIT Process Handbook. The *mapping rules* are the ones partially presented above and fully described in Rajsiri et al. (2010).

5.2. PIM level

The general principle of this second step (described in Touzi et al. 2009) and detailed in Touzi 2007) is the following: *applying a model transformation to deduce from a collaborative process model (respecting the MM described in Section 4.2), a model of an SOA logical architecture (respecting the MM described in Section 5.3*). Obviously, the collaborative process model used to start the model transformation is the one obtained from the business step described in Section 5.1.

The initial question at this point is the following: Is there enough cognitive material inside a BPMN collaborative process to build an SOA architecture model? This point has been discussed in Touzi et al. (2009) on the basis of Touzi, Bénaben and Pingaud (2007) and Booch, Jacobson and Rumbaugh (2004). The conclusion was that, except for the fact that data and services are abstract business entities (coming from the MIT Process Handbook), which should be replaced by real data and services at the implementation time (see PSM level), the knowledge contained inside the process model is sufficient to define the mediator architecture.

Figures 10–12 show graphical representations of the set of rules that are applied during transformation to generate the three views of the SOA model.

Circles located in the middle of two class diagrams represent the rules. The class diagrams are sub-graphs, which are parts of the MMs presented. On the left part of each figure is the sub-graph of the source MM and on the right part is the sub-graph of the target MM. The rules have to be interpreted in the following manner: 'When an object is identified in the collaborative process model (belongs to the left-side sub-graph linked to the rule), it will be transformed into an object instantiated from the class on the right side of the figure. This means that it will become an object in the collaborative information system of the network'.

Rules from Figure 10 can be described in natural language as follows (rules are formally described in Touzi et al. 2009):

- **Rs1**: For each CIS task in the collaborative process model, a CIS service is generated, either specific or generic. An annotation (generic) is added to the process model task to make it easier to identify generic CIS services.
- **Rs2**: The CIS lane of the collaborative process corresponds to an attribute of the collaborative service class, which defines the organisation of services of the CIS according to different categories.
- **Rs3**: This rule is similar to Rs1 but concerns the deduction of a partner service from a partner task.
- **Rs4**: This rule expresses the organisation of the partners' services. An attribute (enterprise division) shows the partner division to which the service belongs.
- **Rs5**: This rule is not a rule to be implemented, but it shows the need for additional knowledge to obtain a complete and useful view of services. This additional knowledge concerns a description of service implementations (address, access protocols etc.).

For instance, in natural language:

- **Ri1**: This rule concerns the data element that is associated with the message flow element. The deduced business object elements refer to the messages (data) exchanged between partners in the collaboration.
- **Ri2**: This rule is not a rule to be implemented, but it shows the limits of the BPMN model in describing exchanged business objects (invoice, order etc.). As

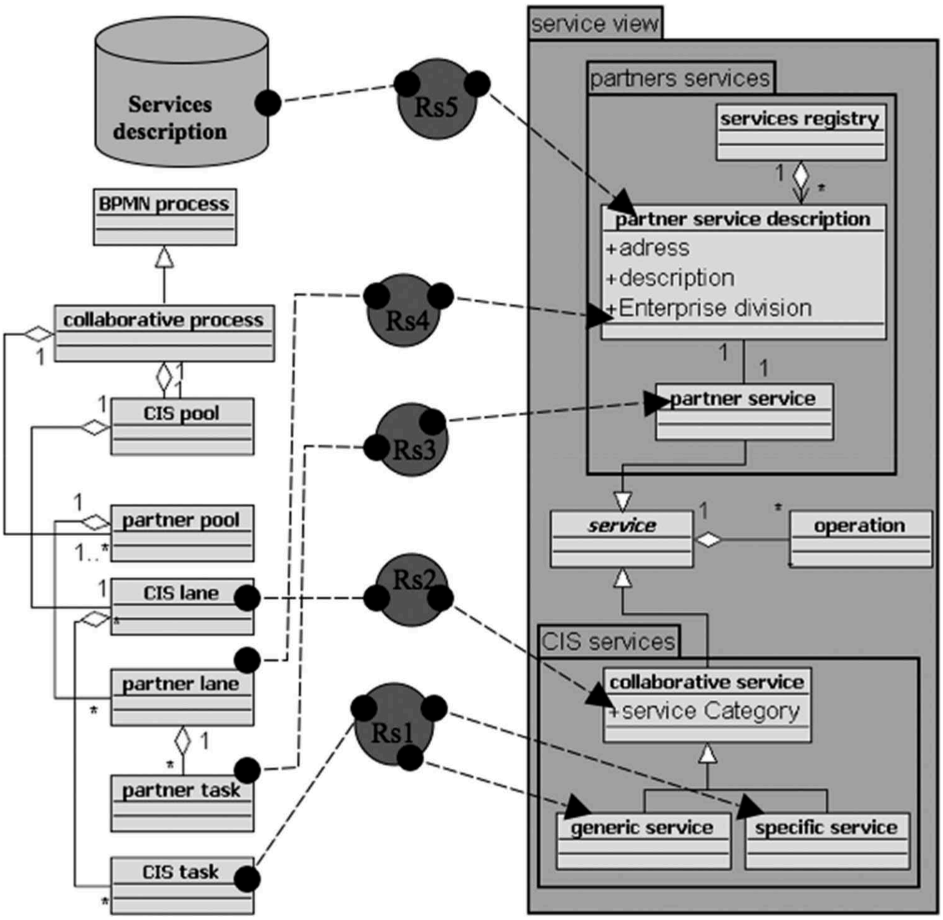

Figure 10. Transformation rules for services view.

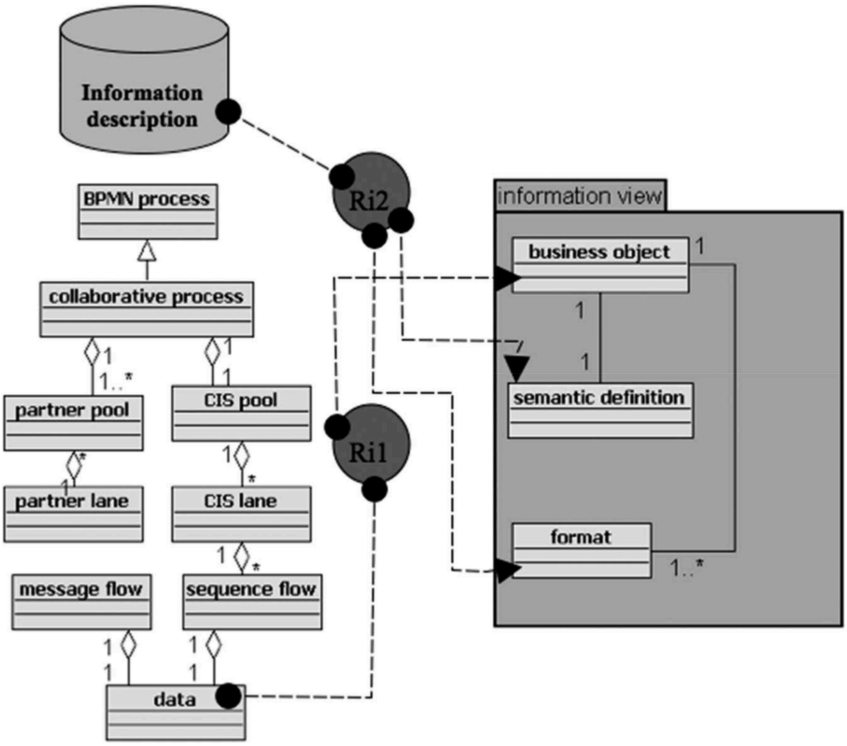

Figure 11. Transformation rules for information view.

previously stated, the transformation is not sufficiently developed in this view. Additional knowledge is needed to describe the structure of information.

For instance, in natural language (partial list of rules, see Touzi et al. 2009 for full description):

- **Rp1**: This rule concerns the deduction of partner element that is important to specify the holder of one activity from the BPMN partner lane element.
- **Rp4**: This rule allows the transformation of BPMN gateways into different BPEL elements (pick, flow and switch) depending on the type of gateway:
  - if it is a parallel gateway, a flow class will be generated to express a parallel execution of activities;
  - if it is a data-based inclusive gateway, a flow class will be generated, associated with a switch class for each set of activities linked to the gateway;
  - if it is an event-based exclusive gateway, a pick class will be generated to express that an event must be produced to continue the execution of the process;
  - if it is a data-based exclusive gateway, a switch class will be generated to express that the continuation of the execution of the process depends on the value of a variable.

The ATL tool, presented in Bézivin et al. (2003), supports the model transformation. Regarding Section 5.1, the *source MM* is the collaborative process MM, the *target MM* is the SOA architecture of the MIS while the *additional knowledge* is the UML SOA profile. The *mapping rules* are described in Touzi et al. (2009).

5.3. PSM level

The general principle of this third step is to apply a model transformation to deduce a model of a technical architecture from a UML model of the SOA architecture of the MIS (respecting the MM described in Section 5.3). This technical architecture model (which respects the MM described in Section 5.4) includes all the knowledge required for an ESB deployment.

The model transformation is implanted by JDOM.[1] Regarding Section 5.1, the *source MM* is the SOA architecture of the MIS, the *target MM* is the technical architecture of the MIS while the *additional knowledge* contains all the technical elements concerning services and data (schematically, this consists in the elements embedded in WSDL files).

The transformation principle uses an intermediate model (technical knowledge model, based on a technical knowledge MM), which embeds the *additional knowledge* (Figure 13).

The following *mapping rules* (Figure 14) allow the logical model to be 'enriched' with the technical knowledge to achieve the technical model:

- Rule 1: Each *service* of the technical knowledge MM maps to a *service* in the technical MM in which the attribute *realname* is added.
- Rule 2: The *bpel* element of the technical knowledge MM maps to *collaborative business process* in the technical MM (including attributes).
- Rule 3: The *wsdl* element of the technical knowledge MM maps to *partner service description* in the technical MM (including attributes).
- Rule 4: Each *operation* of the technical knowledge MM maps to an *operation* in the technical MM.

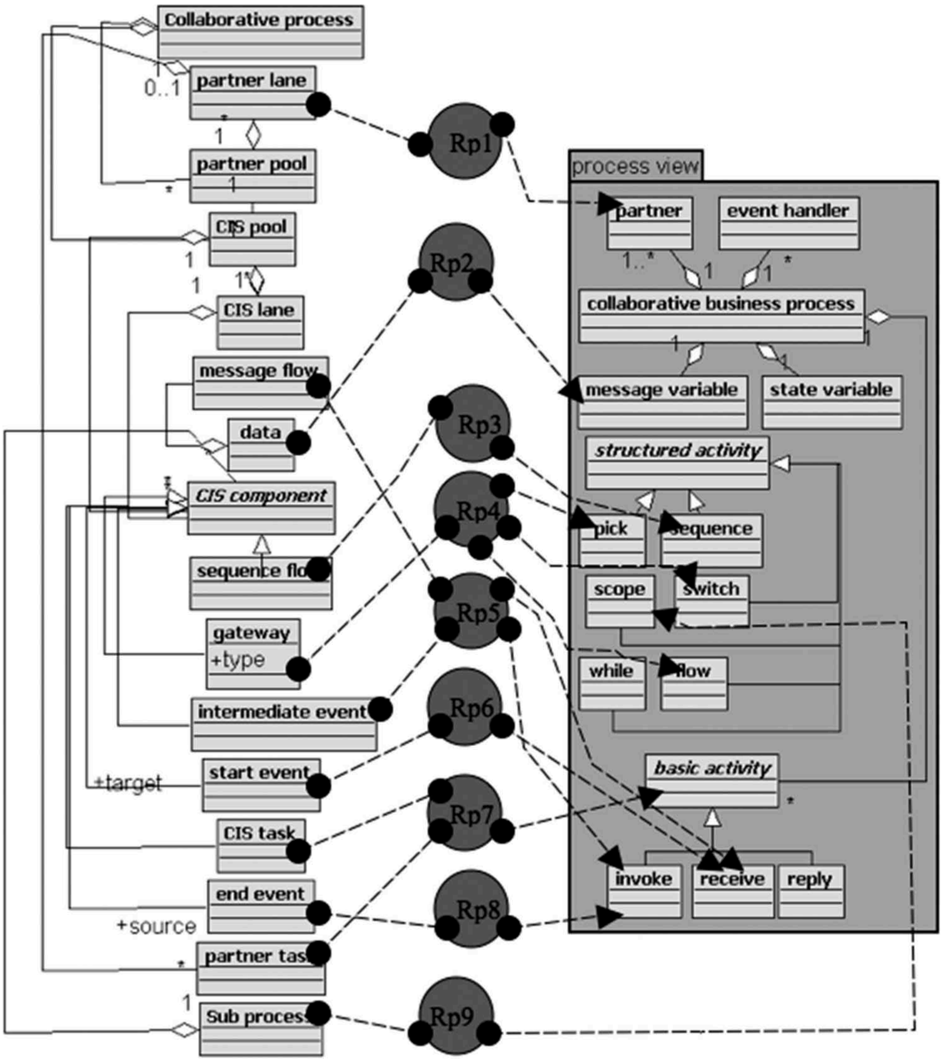

Figure 12. Transformation rules for process view.

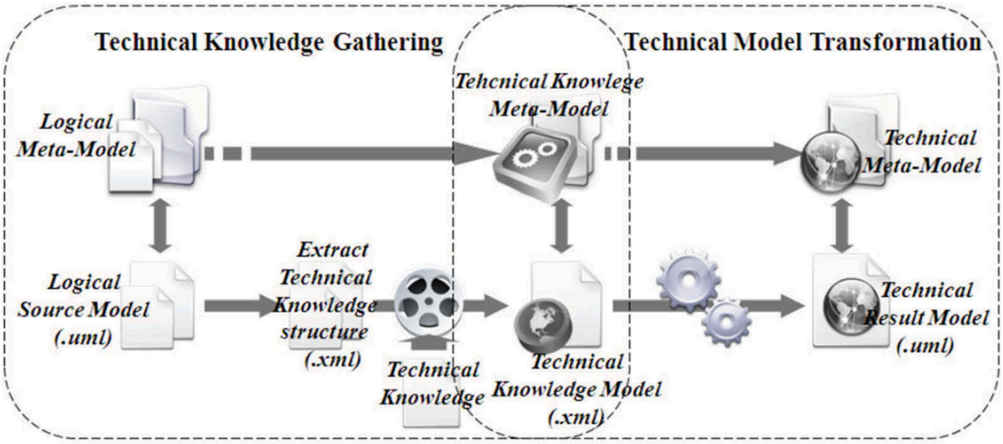
Technical Knowledge Gathering
Technical Model Transformation
Tehcnical Knowlege Meta-Model
Logical Meta-Model
Technical Meta-Model
Logical Source Model (.uml)
Extract Technical Knowledge structure (.xml)
Technical Knowledge
Technical Knowledge Model (.xml)
Technical Result Model (.uml)

Figure 13. Model transformation for the technical model design.

- Rule 5: First, for each *input* of the technical knowledge MM, one attribute *input* is generated in the *operation* element in the technical MM. Second, the attribute *real-name* is added in the *business object* element of the technical MM. Thus, *operation* is linked with *business object* through *input*.
- Rule 6: First, for each *output* of the technical knowledge MM, one attribute *output* is generated in the *operation*

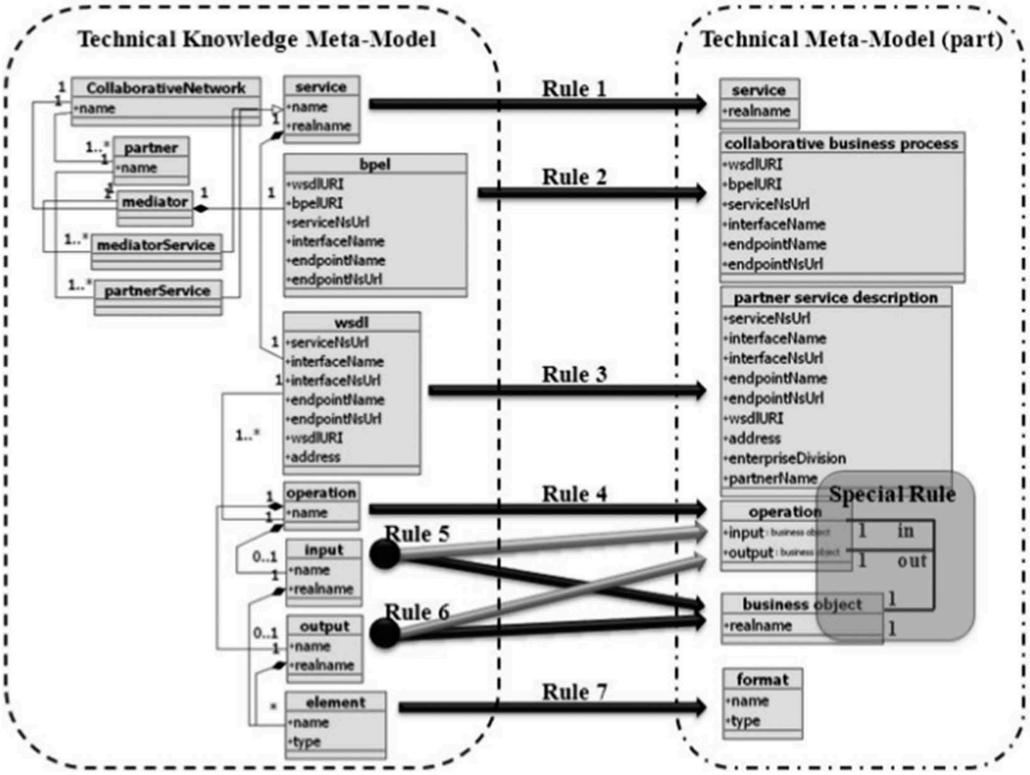

Figure 14. Mapping rules at the PSM level.

element in the technical MM. Second, the attribute *realname* is added in the *business object* element of the technical MM. Thus, *operation* is linked with *business object* through *output*.

- Rule 7: Each attributes *name* and *type* of *element* of the technical knowledge MM maps, respectively, to the attributes *value* and *type* of *format* of the technical MM.
- Special rule: This rule is dedicated to changing the end links of association *in* and *out* from *service* to *operation*.

One important point at this level is that the use of this intermediate technical knowledge model is due to the need for mapping between business services (and data) and real technical services (and data). This mapping is not automatic and is based on manual mapping (Mu 2011).

5.4. Deployment

The general principle of this fourth step is to use the technical model (obtained at the PSM level) and the available technical elements to generate the files described at the end of Section 5.5 (SE, BC, SA and SA-BPEL), required for PEtALS ESB deployment.

First of all, the following list contains all the available technical elements that can be used at this stage:

- PEtALS ESB: the communication middleware based on JBI standard.
- SE-WF: the orchestration service provided with PEtALS ESB, named SE-BPEL.
- BC: some generic BCs, provided with PEtALS ESB, especially BC-SOAP (but also BC-FTP, BC-EJB, BC-SFTP, BC-MAIL).
- WSDL: All the description files provided for any external operational service that will be involved in the collaborative behaviour. It contains not only the specification of input(s) and output(s) of the service, some behavioural description but also the communication protocol required by the service (and thus the relevant BC).

Due to the fact that SE and BC are technical files, available in the ESB, the main problem is to generate the SA files (for services) and the SA-BPEL file (for workflow). The generation principle is shown in next picture (Figure 15):

Each SA file should contain (1) the link to the BC or the SE it will be connected to, and (2) the SU of the service(s) concerned by this BC or SE. The *PSM model* shows the whole set of services (external and internal) involved in the collaborative process. For each service of this set, the associated *WSDL* file provides the communication protocol, and the dedicated *SA* file might also be generated (if not existing already). This *SA* file might then be fulfilled with the connection file (*link BC/SE*) and the *SU* file(s), thanks to the specification (input/output) contained in the WSDL file(s). A specific *java* code is in charge of this whole task.

The SA-BPEL file should contain (1) the link to the SE-WF used for workflow orchestration and (2) the SU-BPEL, i.e. the WSDL files of the services involved in the

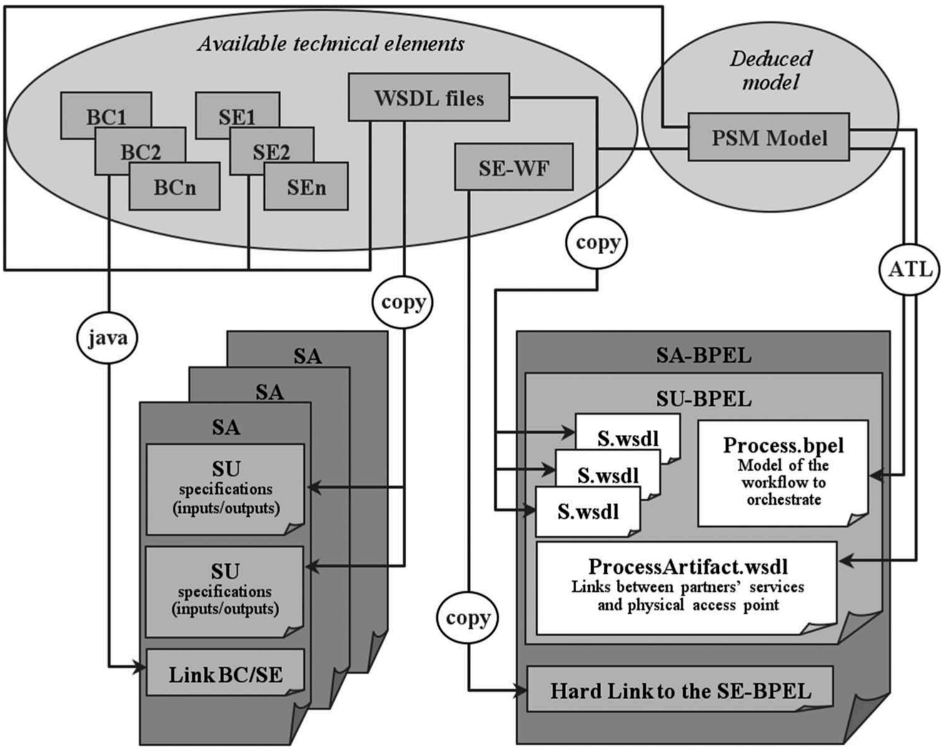
Available technical elements
BC1
BC2
BCn
SE1
SE2
SEn
WSDL files
SE-WF
Deduced model
PSM Model
copy
copy
copy
java
ATL
SA
SA
SA
SU specifications (inputs/outputs)
SU specifications (inputs/outputs)
Link BC/SE
SA-BPEL
SU-BPEL
S.wsdl
S.wsdl
S.wsdl
Process.bpel Model of the workflow to orchestrate
ProcessArtifact.wsdl Links between partners' services and physical access point
Hard Link to the SE-BPEL

Figure 15. From available components and model to deployment artefacts.

collaborative workflow, the process file defining the workflow and the specific file dedicated to linking services and their physical addresses (endpoints). As regards the link to the SE-WF, this is currently a hard connection to the SE-BPEL tool, embedded in the PEtALS ESB. As concerns the SU-BPEL, the *PSM model* shows the whole set of services (external and internal) involved in the collaborative process. For each one of this set of services, the associated *WSDL* file may be copied in the SA-BPEL. An *ATL* transformation extracts from the *PSM model* the elements required to build both the *Processartifact.wsdl* file (connecting services and their endpoints) and the *Process.bpel* file (defining the behaviour).

This final technical step extracts from the PSM model (and the technical components such as WSDL files) the elements required to build all the files, necessary to deploy the PEtALS ESB entirely and efficiently, and specifically dedicated to the collaborative situation. At the end of this stage, as long as the ESB, partners' services, SEs and BCs are available, this set of technological components can be configured in one adequate mediator, running and orchestrating the collaborative behaviour 'interoperably'.

5.5. Agility

According to Kidd (1994), Lindberg (1990) and Sharifi and Zhang (1999), agility may be seen as 'an ability to satisfy a change in a short time'. The main factor of MIS agility is based on the MIS engineering approach and especially the MIS re-engineering approach. Indeed, this design method has been implemented in an ESB where each tool of each step has been connected as a service (i.e. CIM, PIM and PSM tools). Thus, the native plasticity of the workflow orchestration of that platform may be used in order to reconfigure the MIS.

Furthermore, run-time services may be connected on the same ESB (Figure 16). In that case, the mediation cycle is the following: (1) a design-time workflow is orchestrated on the ESB, which invokes design-time services one after the other (CIM, PIM, PSM) in order (2) to deduce MIS PSM model and run-time workflow; (3) run-time workflow is orchestrated on the ESB, which invokes run-time services one after the other in order to perform the collaborative situation; (4) an adaptation requirement is detected during run-time orchestration, which implies invoking the design-time workflow from the most appropriate design-time service. For instance:

- If the collaborative situation changes, then it should be started from the CIM level.
- If the situation has not changed but the partners have (new partner, lack of resources etc.), then it should be started from the PIM level.
- If situation and partners have not changed but any dysfunction occurs during run time (one run-time service did not provide the expected result), then it should be started from the PSM level.

Finally, the above considerations present how the SOA provides a merge between design time and run time.

6. Illustration

This section is dedicated to introduce a simple example of collaborative situation in the manufacturing domain. A customer (*Client*) and two providers (*Fournisseur 1* and *Fournisseur 2*)

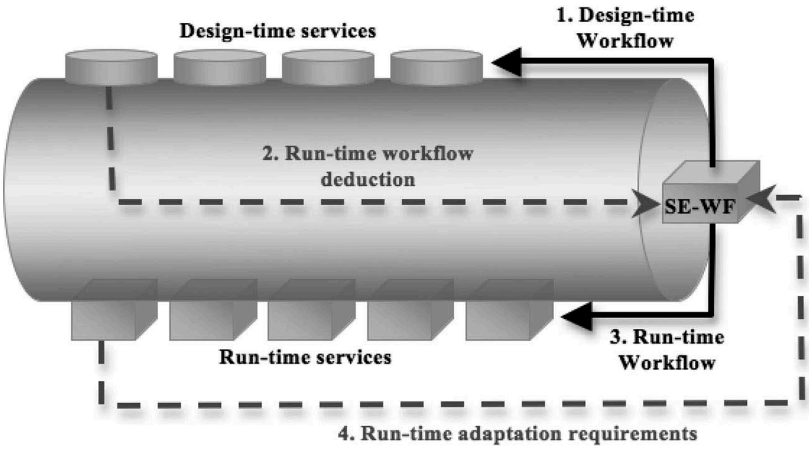

Figure 16. Agile structure of MIS.

are using a MIS (CIS) to collaborate for the manufacturing of a product. This example focuses on the *business* to *logical* transformation. The steps above (deducing the collaboration business process) and beyond (deploying the actual MIS) have been presented, respectively, in Rajsiri et al. (2010) and Touzi et al. (2009). The following BPMN model (Figure 17) describes the collaboration business process (i.e. the *business* level):

This BPMN model (more precisely, the XML file of this BPMN diagram) is injected into the transformation service. By applying the transformation rules presented in Section 5, the following UML model is obtained (Figure 18):

This logical model, describing the mediation IS dedicated to support the collaboration between the customer and its two providers, is incomplete (there are too many classes to be presented on this diagram. For instance, the following picture illustrates specifically the package 'partner services' in the 'service view' (Figure 19).

Finally, this example shows, in a manufacturing context, that the MISE approach could efficiently support emerging collaboration by providing the collaborative network (in that case one customer and two providers) with the interoperable infrastructure able to invoke relevantly their services (the hypothesis of SOA compliant members of the collaborative network is still pending) to perform efficiently their collaboration and reach the common goals.

Considering some other existing approach or tools (research results or industrial products), Table 1 describes the different features of these existing 'competitors' with regards

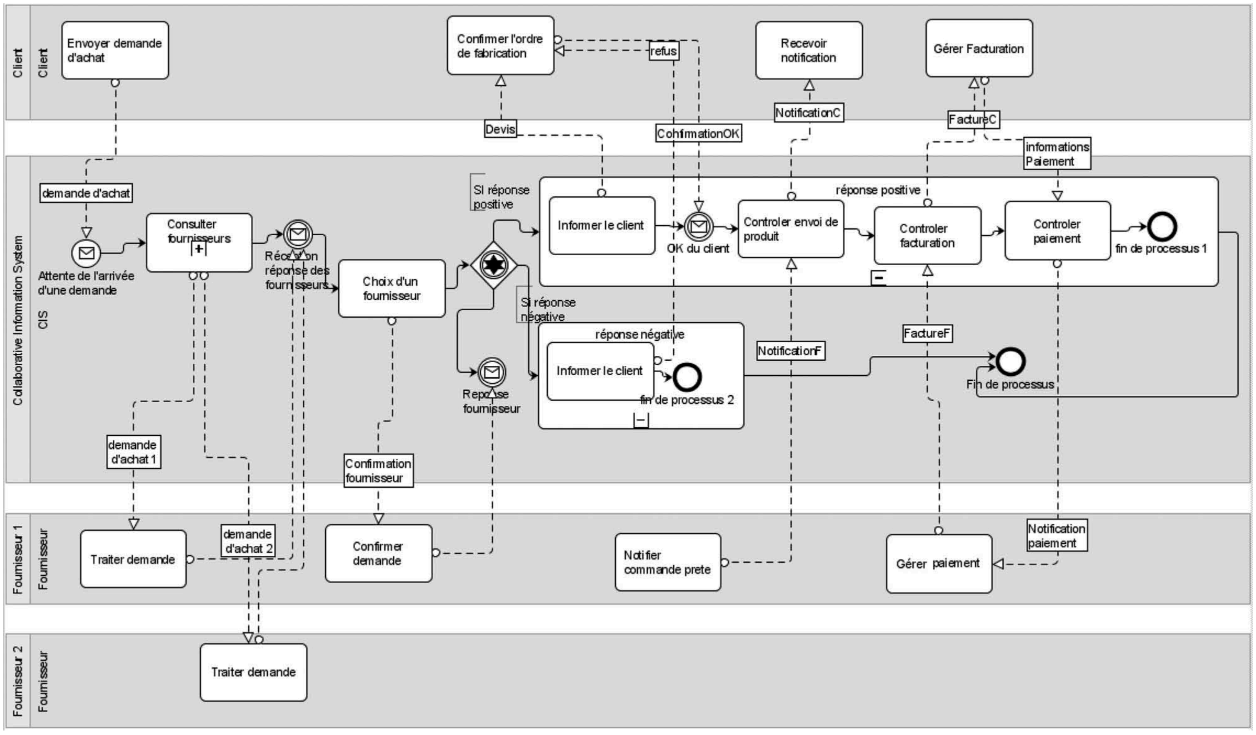

Figure 17. BPMN example (CIM model) in a manufacturing context.

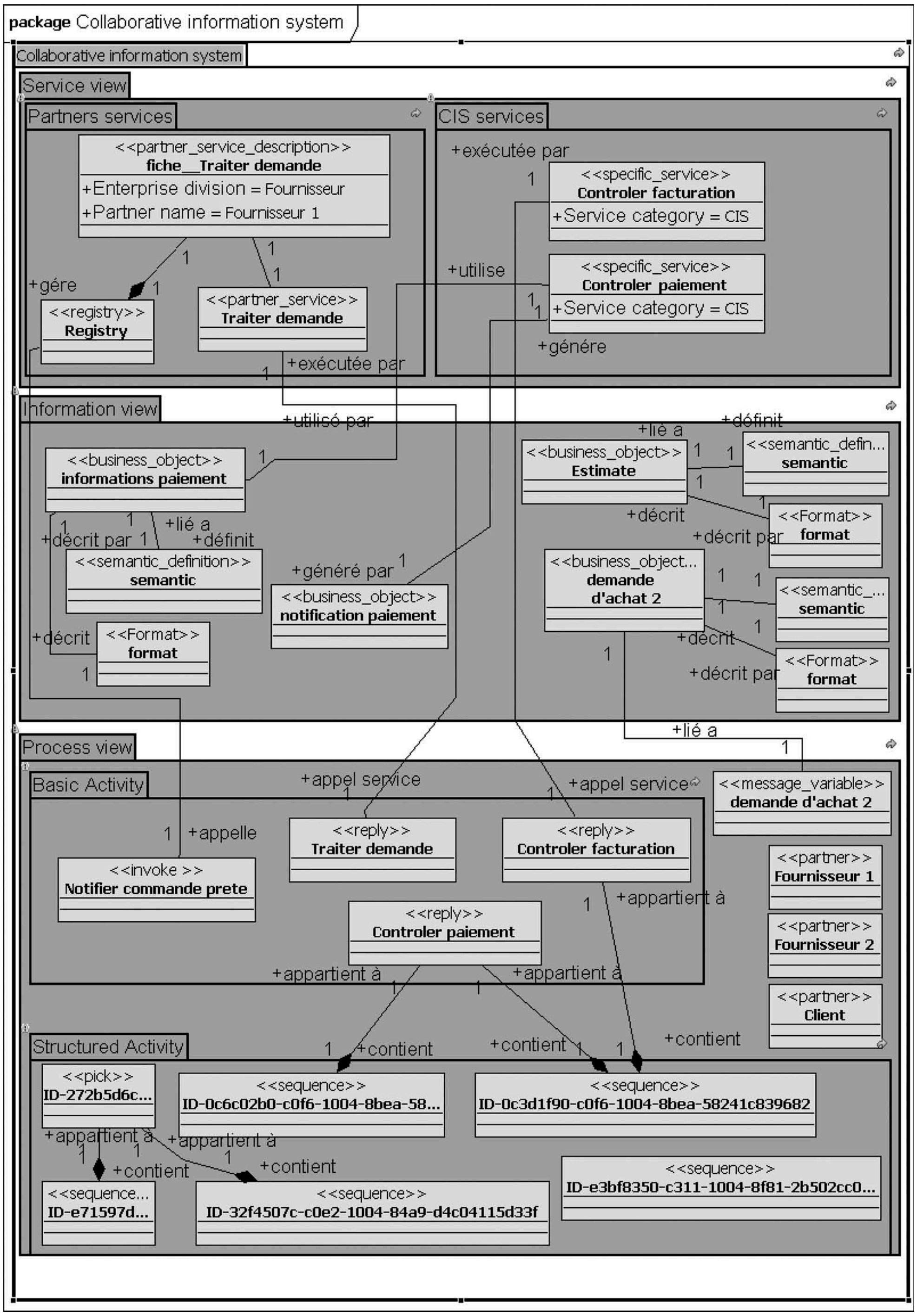

Figure 18. UML example (PIM model) in a manufacturing context.

to the four levels (1) business, (2) logical, (3) technological and (4) agility. Table 1 presents two existing products providing similar features (Bizagi and Run my Process) and four research results presenting interesting perspectives. For each of them, the bold text describes the main difference with regards to the MISE approach.

Obviously, the Table 1 shows that there are very actual perspectives to the MISE approach in order to shade its weaknesses. Besides, the table also shows that there are actual products available on the market that provides some similar features. However, all these products underline the real contributions of MISE (i.e. the ones that are not available on these products): (1) automated business process design, (2) connection with legacy IS (through web services) and (3) agility management. The fact that these features have not been implemented in these products shows that either they are useless or it is real contribution. This paper claims that these features are not useless. Finally, Table 1, on the one hand,

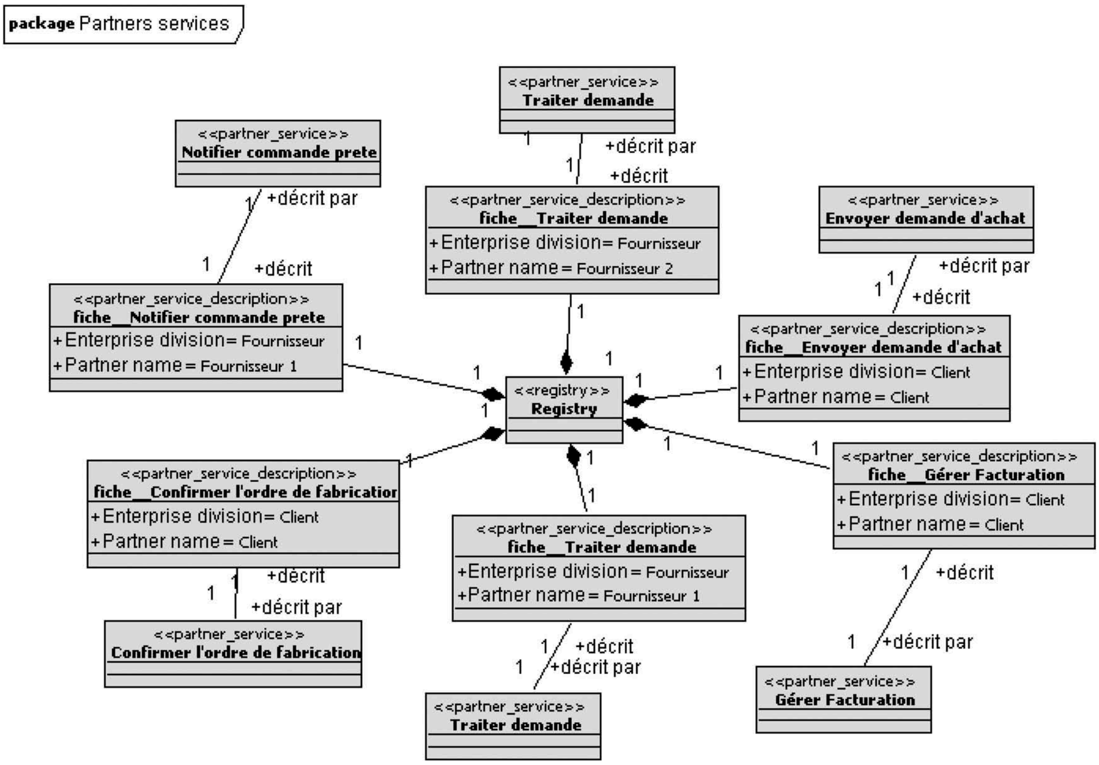
package Partners services
<<partner_service>>
Traiter demande
+décrit par
+décrit
<<partner_service>>
Notifier commande prete
+décrit par
+décrit
<<partner_service_description>>
fiche__Traiter demande
+Enterprise division= Fournisseur
+Partner name= Fournisseur 2
<<partner_service>>
Envoyer demande d'achat
+décrit par
+décrit
<<partner_service_description>>
fiche__Notifier commande prete
+Enterprise division= Fournisseur
+Partner name= Fournisseur 1
<<partner_service_description>>
fiche__Envoyer demande d'achat
+Enterprise division= Client
+Partner name= Client
<<registry>>
Registry
<<partner_service_description>>
fiche__Gérer Facturation
+Enterprise division= Client
+Partner name= Client
<<partner_service_description>>
fiche__Confirmer l'ordre de fabricatior
+Enterprise division= Client
+Partner name= Client
+décrit
+décrit par
<<partner_service>>
Confirmer l'ordre de fabrication
<<partner_service_description>>
fiche__Traiter demande
+Enterprise division= Fournisseur
+Partner name= Fournisseur 1
+décrit
+décrit par
<<partner_service>>
Traiter demande
+décrit
+décrit par
<<partner_service>>
Gérer Facturation
1

Figure 19. Detailed part of the package 'partner services'.

shows clearly the main contributions of the MISE approach and, on the other hand, draws the lines for perspectives that will be presented and discussed in the conclusion.

7. Conclusion

This paper presents an approach to designing the global MIS of an emerging collaborative situation. This approach is covering the whole path from the early stages of design time (partners' characterisation and description of the collaborative objectives) to the late stages of runtime (actual implementation and constant updating thanks to agility management). This global IS is based on existing ISs (from involved organisations) and on a MIS providing interoperability to the ISs. The whole approach is model driven and uses several model transformations from business layer to technical layer. Each of these transformations uses mapping rules between source and target MMs to build a target model from a source model and some additional knowledge. Finally, using the PSM, deployment is possible on the target ESB platform. It is also interesting to notice that the granularity question can be managed by the MISE approach. Actually, there might be several levels of collaboration, especially in manufacturing context where there are strong connections between very different behavioural layers such as business and decision layers, production and assembly layers, supply chain and logistics layers, human resources layers etc. Therefore, there may be an MIS, managing a collaboration involving several MISs. This 'fractal' feature is not due to the contributions presented in this paper but to the conjunction of using a BPM approach on an SOA architecture.

Although this paper presents an overview of a deduction approach for MIS design, there are four main restrictions, which lead to four perspectives for improvement:

- Each partner of the emerging collaborative situation is supposed to be able to provide its own services (SOA), which is not realistic. Hence, one of the main next steps concerns working on service design (and service-oriented user interface principles) to help partners to convert their abilities (even human-based abilities) into services or to connect them to dedicated interface services. This part of perspective may be solved by agents (Papakostas et al. 2016), fuzzy logic (Renna 2016; Gholamian, Mahdavi, and Tavakkoli-Moghaddam 2016) or (Mas et al. 2016).
- The deduced process (at the business level) is entirely based on the MIT Process Handbook concepts and, as such, is mainly *operational* (in the sense of the ISO 9001 standard). There is no deduction concerning *decisional* and *support* processes. Still, an emerging collaborative situation should probably follow the same behavioural rules as an organisation and should thus follow that standard. The CIM level should focus on providing collaborative process cartography (instead of one collaborative process). The work presented in Mu, Bénaben and Pingaud (2015) could be integrated in the business layer to tackle this issue.
- The mapping between business services and technical services is 'hand made'. This is not an easy question: it is not a one-to-one mapping but a global mapping between the set of business services (and data) and the set of technical services (and data). Semantic recovery (or semantic reconciliation) for this business-to-technical mapping of services and data is also one critical improvement to consider.

Table 1. Comparison with existing results.

	Business	Logic	Technology	Agility
MISE	This approach provides a way to automatically deduce a business process model based on the characterisation of partners' capabilities and collaborative objectives	A UML intermediary model of the MIS is built by model transformation of the collaborative business process model. The required web services of partners are automatically deduced from expected business tasks	An executable workflow file is generated automatically from the UML MIS model and is implemented through PEtALS ESB	Detecting if the execution of the current collaborative workflow should be interrupt is manual. Deciding if the design-time process should be restarted from CIM, PIM or PSM level is manual. Deployment of the new MIS (whatever the starting level) is automated
Barthe et al. (2014)	No	No	No	This result provides a way to automatically monitor de process and to detect if anything is different from the expected situation. **It provides an automated implementation of the mechanisms detailed in this paper**
Bizagi	This software (first launched in 1989) allows a local modelling of BPMN models (using ECLIPSE IDE). **The modelling is 100% manual**	No	The obtained process can be locally executed as a workflow. **There is no connection with existing web services or information system**	No
Boissel Dallier et al. (**2015**)	No	This research contribution provides an approach to select automatically technical web services (described using SAWSDL) to perform business tasks of a business process model. This is a one-to-many and somehow many-to-many matching mechanism. **It provides a deeper and better matching between business tasks and actual web services**	The obtained workflow is directly executable by a classical workflow engine (e.g. BPEL engine or ACTIVITI). **The orchestration is very similar to the MISE approach**	No
Mu, Bénaben, and Pingaud (2015)	This research work provides an approach dedicated to build the whole process cartography of a collaborative network (including decisional, operational and support processes) **It provides a wider solution for the business layer (based on very similar principles)**	No	No	No
Run My Process	This fully online tool provides a SaaS business process designer. **The modelling is 100% manual**	Existing applications and web services from legacy information systems can be connected. **This requires a manual configuration but this is a fair connection of existing services**	The obtained workflows are executed on a SaaS workflow engine. **The orchestration is very close to the one provided by the MISE approach**	No

Consequently, the research results presented in Boissel-Dallier et al. (2015) and based on Bénaben et al. (2010), Bénaben et al. (2013) and Sivashanmugam et al. (2005), dealing with the gap between, on the one hand, processes, functions and information and, on the other hand, workflows, services and data, could also be integrated into the whole approach.

- Finally, agility may be seen as the combination of *detection* and *adaptation*. However, the SOA features presented, which provide an appropriate way to invoke design-time services, are especially dedicated to managing *adaptation* aspects. But what about *detection*? Event-driven architecture and complex event processing are very promising tools to deal with information emitted from run-time services in order to detect any unexpected evolution. The research works presented in Barthe et al. (2014) could also help to deal with this issue.

The approach presented in this paper is not dedicated to be implemented as it has been described in the previous pages. Actually, even if the approach can be considered as the result of a very applied research work, some steps of the approach (design of models, populating of the knowledge base etc.) present some operational challenges. However, the presented approach demonstrates the continuous path that this approach draws between the very high business layers (descriptions of enterprises) and the very low technological layers (deployment of an operational MIS).

Note

1. JDOM provides a means of reading and writing XML data. http://www.jdom.org/mission/index.htm.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Frederick Benaben http://orcid.org/0000-0002-1676-6524

References

Agostinho, C., Y. Ducq, G. Zacharewicz, J. Sarraipa, F. Lampathaki, R. Poler, and R. Jardim-Goncalves. 2016. “Towards a Sustainable Interoperability in Networked Enterprise Information Systems: Trends of Knowledge and Model-Driven Technology.” In *Computers in Industry*. Vol. 79, 64–76. Elsevier.

Anaya, V., G. Berio, M. Harzallah, P. Heymans, R. Matulevicius, A. L. Opdahl, H. Panetto, and M. Jose Verdecho. 2010. “The Unified Enterprise Modelling Language-Overview and Further Work.” *Computers in Industry* 61: 99–111. doi:10.1016/j.compind.2009.10.013.

Aubert, B., and A. Dussart, “Inter-Organizational Information Systems.”. Bourgogne Report. CIRANO, 2002.

Barthe, A.-M., S. Truptil, F. Bénaben, and H. Pingaud. 2014. “Event-Driven Agility of Interoperability during the Run-Time of Collaborative Processes.” In *Decision Support System*. Vol. 59, 171–179. Elsevier.

Bénaben, F., N. Boissel-Dallier, J.-P. Lorré, and H. Pingaud. 2010. “Semantic Reconciliation in Interoperability Management through Model-Driven Approach.” In *Proceedings of Pro-VE’10*, 705–712. Vol. 336. Saint-Etienne, France: Springer.

Bénaben, F., N. Boissel-Dallier, H. Pingaud, and J.-P. Lorré. 2013. “Semantic Issues in Model-Driven Management of Information System Interoperability.” *International Journal of Computer Integrated Manufacturing* [Taylor&Francis.] 26: 1042–1053. doi:10.1080/0951192X.2012.684712.

Bénaben, F., J. Touzi, V. Rajsiri, and H. Pingaud. 2006. “Collaborative Information System Design.” In *IS and Collaboration: State of the Art and Perspectives, GI-Edition, Lecture Notes in Informatics*, 281–296.

Benguria, G., X. Larrucea, B. Elvaester, T. Neple, A. Beardsmore, and M. Friess. 2006. “A Platform Independent Model for Service Oriented Architectures.” In Proceedings of InterOp-ESA’06, 407–416. Bordeaux, France: Springer.

Bézivin, J., G. Dupé, F. Jouault, G. Pitette, and J.-E. Rougui. 2003. “First Experiments with the ATL Model Transformation Language: Transforming XSLT into Wquery.” In Proceedings of OOPSLA’03. Anaheim, USA. Association for Computing Machinery Press (ACM SIGPLAN notices:38).

Boissel-Dallier, N., F. Bénaben, H. Pingaud, and J.-P. Lorré. 2015. “Mediation Information System Engineering Based on Hybrid Service Composition Mechanism.” *Journal of Systems and Software (JSS)* Elsevier. 108: 39–59. doi:10.1016/j.jss.2015.05.064.

Booch, G., I. Jacobson, and J. Rumbaugh. 2004. *UML 2.0 Reference Manual*. CampusPress.

Estevez, E., A. Sanchez-Garcia, J. Gamez-Garcia, J. Gomez-Ortega, and S. Satorres-Martinez. 2016. “A Novel Model-Driven Approach to Support Development Cycle of Robotic Systems.” *International Journal of Advanced Manufacturing Technology* Springer. 82-1: 737–751. doi:10.1007/s00170-015-7396-4.

Gao, X., X. Wang, Y. Li, M. Yang, Y. Liu, and W. Guo. 2016. “Workflow Dynamic Change and Instance Migration Approach Based on Polychromatic Sets Theory.” *International Journal of Computer Integrated Manufacturing* Taylor&Francis. 29: 386–405. doi:10.1080/0951192X.2015.1034181.

Gholamian, N., I. Mahdavi, and R. Tavakkoli-Moghaddam. 2016. “Multi-Objective Multi-Product Multi-Site Aggregate Production Planning in a Supply Chain under Uncertainty: Fuzzy Multi-Objective Optimisation.” *International Journal of Computer Integrated Manufacturing* Taylor&Francis. 29-2: 149–165.

Kent, S. 2002. “Model-Driven Engineering.” In Proceedings of IFM’02. pp. 286-298, Springer Berlin/Heidelberg. Turku, Finland.

Kidd, T. P. 1994. *Agile Manufacturing: Forging New Frontier*. Addison-Wesley. Series in Manufacturing Systems.

Konstantas, D., J.-P. Bourrières, M. Léonard, and N. Boudjlida. 2005. *Interoperability of Enterprise Software and Applications*. Genova, Switzerland: Springer.

Lindberg, P. 1990. “Strategic Manufacturing Management: A Proactive Approach.” *International Journal of Operation and Production Management* Emerald. 10: 94–106. doi:10.1108/01443579010001221.

Lundgren, M., M. Hedlind, and T. Kjellberg. 2016. “Model Driven Manufacturing Process Design and Managing Quality.” In Proceedings of 26th CIRP Design Conference, 299–304. Amsterdam, Netherland: Elsevier Science Bv.

Maier, M. W. 1998. “Architecting Principles for Systems-Of-Systems.” *Systems Engineering* 1: 267–284. doi:10.1002/(ISSN)1520-6858.

Malone, T. W., K. Crowston, and G. A. Herman. 2003. *Organizing Business Knowledge – the MIT Process Handbook*. ISBN 0-262-13429-2. MIT Press. Chapters 1. 3.

Mas, F., J. Rios, A. Gomez, and J. C. Hernandez. 2016. “Knowledge-Based Application to Define Aircraft Final Assembly Lines at the Industrialisation Conceptual Design Phase.” *International Journal of Computer Integrated Manufacturing* Taylor&Francis. 29: 677–691. doi:10.1080/0951192X.2015.1068453.

Morley, C., J. Hugues, and B. Leblanc. 2002. *UML pour l’analyse d’un système d’information*. 2nd ed. Dunod, France.

Mu, W. 2011 “Model Transformation Research from Logical to Technical Model in MISE Project.” internal report, unpublished.

Mu, W., F. Bénaben, and H. Pingaud. 2015. “Collaborative Business Process Deduction in a Model-Driven BPM Approach.” In *Enterprise Information System*. vols. 9-4, 349–383. Taylor&Francis.

Obermeier, M., S. Braun, and B. Vogel-Heuser. 2015. “A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability.” *IEEE Transactions on Industrial Informatics* 11: 790–800. doi:10.1109/TII.2014.2346133.

OMG. 2003. *MDA Guide Version 1.0.1, Omg/2003-06-01*. Available online.

Papakostas, N., G. Pintzos, C. Giannoulis, and G. Chryssolouris. 2016. “An Agent-Based Collaborative Platform for the Design of Assembly Lines.” *International Journal of Computer Integrated Manufacturing* [Taylor&Francis.] 29: 374–385. doi:10.1080/0951192X.2015.1066862.

Petersen, S. A. 2005. “The Role of Enterprise Modelling in Virtual Enterprises.” *Collaborative Networks and Their Breeding Environments* Springer. 186: 109–116.

Pingaud, H. 2009. *Prospective de recherches en interopérabilité: vers un art de la médiation?* Tarbes, France: Plenary lecture, CIGI’09.

Rajsiri, V. 2009. “Knowledge-based system for collaborative process specification.” PhD thesis, Toulouse University.

Rajsiri, V., J.-P. Lorré, F. Bénaben, and H. Pingaud. 2010. “Knowledge-Based System for Collaborative Process Specification.” In *Computers in Industry*. vols. 61-2, 161–175. Elsevier.

Renna, P. 2016. “Dynamic Pricing of Excess Capacity in Production Networks by Fuzzy Logic.” *International Journal of Computer Integrated Manufacturing* Taylor&Francis. 29: 611–621. doi:10.1080/0951192X.2015.1109142.

Sharifi, H., and Z. Zhang. 1999. “A Methodology for Achieving Agility in Manufacturing Organizations: An Introduction.” *International Journal of Production Economics* Elsevier. 62: 7–22. doi:10.1016/S0925-5273(98)00217-5.

Sivashanmugam, K., J. A. Miller, A. P. Sheth, and K. Verma. 2005. “Framework for Semantic Web Process Composition.” *International Journal of Electronic Commerce* 9-2: 71–106.

Tchoffa, D., N. Figay, P. Ghodous, E. Exposito, L. Kermad, T. Vosgien, and A. El Mhamedi. 2016. “Digital Factory System for Dynamic Manufacturing Network Supporting Networked Collaborative Product Development.” *Data & Knowledge Engineering* Elsevier. 105: 130–154. doi:10.1016/j.datak.2016.02.004.

Touzi, J. 2007. “Aide à la conception de système d’information collaboratif support de l’interopérabilité des entreprises”. PhD thesis, Toulouse University.

Touzi, J., F. Bénaben, and H. Pingaud. 2007. “Model Transformation of Collaborative Business Process into Mediation Information System.” In Proceedings of IFAC’08. Seoul, Korea. Elsevier.

Touzi, J., F. Bénaben, H. Pingaud, and J.-P. Lorré. 2009. “A Model-Driven Approach for Collaborative Service-Oriented Architecture Design.” *International Journal of Production Economics* Elsevier. 121: 5–20. doi:10.1016/j.ijpe.2008.09.019.

Vernadat, F. 1996. *Enterprise Modeling and Integration*. Chapman&Hal.

Wiederhold, G., and M. Genesereth. 1997. “The Conceptual Basis for Mediation Services.” *IEEE Expert: Intelligent Systems and Their Applications* 12: 38–47. doi:10.1109/64.621227.