



**HAL**  
open science

# Prototype to support morphism between BPMN collaborative process model and collaborative SOA architecture model

Jihed Touzi, Frederick Benaben, Hervé Pingaud

## ► To cite this version:

Jihed Touzi, Frederick Benaben, Hervé Pingaud. Prototype to support morphism between BPMN collaborative process model and collaborative SOA architecture model. Enterprise Interoperability III: New Challenges and Industrial Approaches, 2008, Berlin, Germany. pp.145-157, 10.1007/978-1-84800-221-0\_12 . hal-01596381

**HAL Id: hal-01596381**

**<https://hal.science/hal-01596381>**

Submitted on 20 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Prototype to Support Morphism between BPMN Collaborative Process Model and Collaborative SOA Architecture Model

Jihed Touzi, Frederick Bénaben, Hervé Pingaud  
Centre de Génie Industriel, Ecole des Mines d'Albi-Carmaux  
Route de Teillet 81013 ALBI Cedex 9  
{Jihed.touzi, Frederick.benaben, Herve.pingaud}@enstimac.fr

**Abstract.** In a collaborative context, the integration of industrial partners deeply depends of the ability to use a collaborative architecture to interact efficiently. In this paper, we propose to tackle this point according to the fact that partners of the collaboration respect SOA (Service-Oriented Approach) concepts. We propose to design such a collaborative SOA architecture according to MDA (model-Driven Approach) principles. We aim at using business model (the needs) to design a logical model of a solution (logical architecture). The business model is a collaborative business model (in BPMN, at the CIM level), while the logical model is a collaborative architecture model (using UML, at the PIM level). This paper presents the theoretical aspects of this subject, the mechanisms of morphism and the dedicated translation rules. Finally, we show the prototype of a demonstration tool embedding the transformation rules and running those principles.

**Keywords:** transformation, ATL prototype, collaborative process, meta-model, morphism.

## 1 Introduction

The application of model-driven development facilitates faster and more flexible integration by separating system description to different levels of abstraction. The global MDA approach shows that is possible to separate concerns by splitting implementation choices from specifications of business needs. Specifically, the Model Driven Interoperability (MDI) paradigm [1][2][3] proposes to start from an Enterprise Modelling level that means at Computation Independent Model (CIM) level, defining the collaboration needs of a set of partners and to reach a Platform Independent Model (PIM) level defining a logical architecture of a collaborative solution. Finally the Platform specific Model (PSM) can be generated. The three models are in closed connections and passing from one layer to another one, must

be facilitated by vertical transformation rules. Previous research works have shown the benefits of this new paradigm. The PIM4SOA project [4] defines a PIM meta-model based on SOA. In this work, it is possible to generate a PIM model from a Processes, Organization, Product, \* (POP\*) model. The weak point of this work is that there is no description of the **needed morphism** between the CIM model and the PIM model. Other research works like [5] and [6] focus principally on the identification of the two meta-models (CIM and PIM). The Morphism (which contains the definition of the transformation rules) is missed or shortly described.

In this paper we intend to identify a morphism between a collaborative process model and a collaborative SOA architecture. In our PhD work [3] we are actually developing a prototype that transforms a BPMN collaborative process into a collaborative SOA architecture. The generated UML model can be enriched by an additional knowledge about the collaboration (services description, exchanged messages details, etc.).

This paper is organized as follows. Section 2 defines morphism between two models, while section 3 defines the needed meta-models of the BPMN-Collaborative SOA architecture transformation. Transformation rules are illustrated in section 4 and finally we present in section 5 the architecture of the developed prototype to support the defined morphism.

## 2 Definition of a Morphism

We aim to establish a morphism between a BPMN collaborative process model and a collaborative SOA architecture model. Fig 1 shows how to represent schematically the notion of morphism:  $A$ ,  $B$  represent respectively source and target model and  $M$  is the morphism.

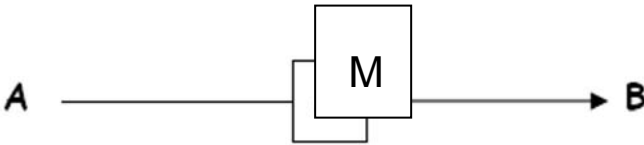


Fig. 1. Morphism between A and B models

Morphism allows obtaining a model B from a model A. It is based on the concepts of *mapping* and *transformation* [7]. If we consider that one model is composed of a set of elements:

- *Mapping* is a *relation* that aims to establish correspondences between elements of two models without modification. The definition of one mapping needs the availability of the two models. The establishment of one mapping needs firstly to define the meta-models that define the models.

- Transformation is a *function* that transforms (or modifies) one source model to a target model. The source model is unchanged and there is generation of new model: the result of the transformation.

Mappings can be of different types [7]: A 1-to-1 mapping puts in correspondence one element of a model with exactly one element of the other model. However, there are other cases in which we map single elements in the source model with a sub-graph in the second model (1-to-n) or, even, a sub-graph to sub-graph (m-to-n).

### 3 Definition of the Meta-Models

In this section, we define the needed meta-models of the BPMN-Collaborative architecture transformation.

#### 3.1 Definition of BPMN Collaborative Process Meta-Model

The first meta-model is of the collaborative process. The meta-model of Fig 2 regroups basic BPMN elements (like gateway, events, message flow...) and other specialized components (like pools or tasks that explicitly refer to collaboration entities). The BPMN formalism aims to support process management for technical and business stakeholders by providing a graphical notation that is rather intuitive and able to represent complex process semantics. As a specialized element the Collaborative Information System (CIS) pool refers to a mediation actor of the collaboration that offers a set of added value services. As example: choosing a supplier from a list of suppliers concerned with a customer order, checking payment transactions, etc. The defined meta-model respects our vision of the collaboration, based on the use of a mediator that facilitates interoperability issues between partners.

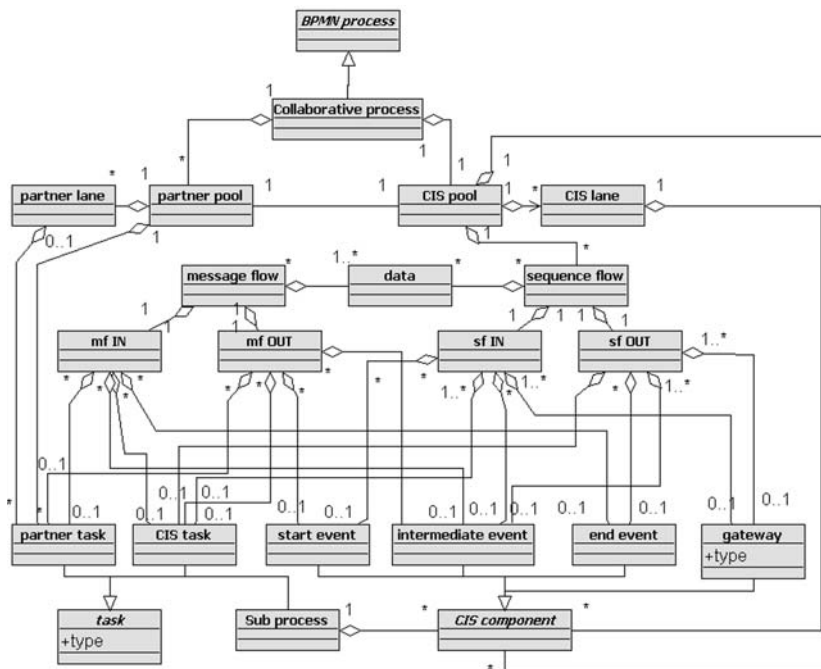


Fig. 2. Collaborative process meta-model (source meta model)

### 3.2 Definition of Collaborative SOA Architecture Meta-Model

The collaborative SOA architecture meta-model described in Fig. 3 is closed and inspired from the PIM4SOA meta-model [4]. Three packages are proposed corresponding to three views:

- **Services view:** services that are used in the collaboration are described; they are business reachable computing functionalities with a known location on the communication network.
- **Information view:** data are exchanged by messages between services, they are defined here in their structure by a data model, and also as a communication utility by identification of the emission and reception services
- **Process view:** interaction amongst services and coordination aspects are specified by the control of processes described here

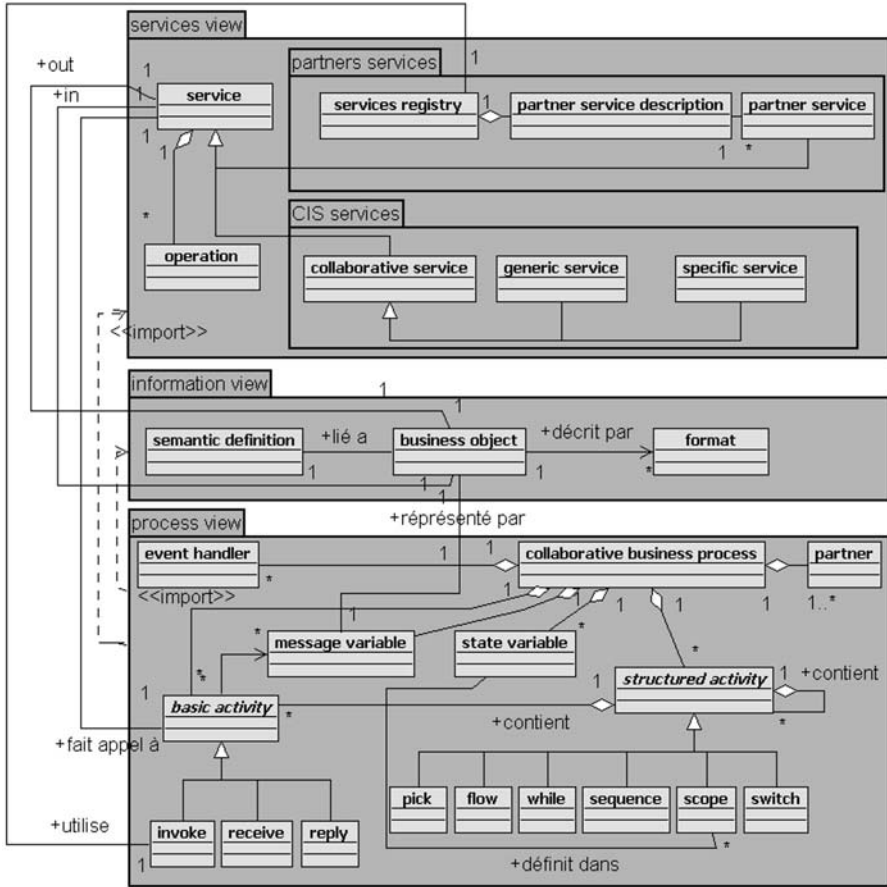


Fig. 3. Meta model of the SOA collaborative architecture (target meta model)

#### 4 Definition of the Transformation Rules

The described source and target meta-models of the morphism are further detailed and justified in [3]. It is not possible to explain here, due to limited space constraints, the objects and the relations that rely them in each meta-model. However, we hope that the following dedicated to the mappings of those meta-models that lead to the transformation rules will give some lights about the equivalences that are used.

Transformation rules are classified in two categories:

1. **Basic generation rules** are used in a first time to create elements of the target model. Most of these rules are defined by a direct mapping between meta model elements.

2. **Binding rules** are applied in a second time to draw the links between the elements resulting from the previous phase. Existing relations in the source model are transformed into relations in the target model.

#### 4.1 Basic Generation Rules

Fig. 4, Fig. 5 and Fig. 6 try to summarise the set of rules (also called derivation laws) that are applied during transformation. The rules are represented by circles located in the middle of two class diagrams. The class diagrams are subgraphs which are parts of the primitive meta models. On the left part of each figure is the subgraph of the source meta model, and on the right part is the subgraph of the target meta model. The rules have to be interpreted in the following manner: *“When an object is identified in the collaborative process model, it belongs to meta-model class of the left side subgraph linked to the rule. Then, it will be transformed in an object instantiated from the class of right side of the figure. We mean that it will become an object in the SOA collaborative architecture”*.

The service view of the SOA collaborative architecture is represented in Fig 4. On the left part, the pool and lane classes are mapped on the different entities services of the right part (partners or CIS services). *Rs1* rule gives the links from tasks in the collaborative process model to services listed in the registries, either specific to the collaboration or generic ones. *Rs2* to *Rs4* rules provide solutions for the structure and organisation of services. *Rs5* shows the need for an additional knowledge to fill services description.

With the same logic, Fig. 5 introduces two transformation rules applied for the information view. As indicated before, the transformation is not sufficiently developed in this domain (business process is not the good way to modelize informations of the collaboration). Transformation provides syntactic indications that help to create business objects (Rules *Ri1* and part of *Ri2*). However, the problem of translation refers to semantic interpretation that we do not include in this part of the study (Remaining part of *Ri2* is probably not a robust solution).

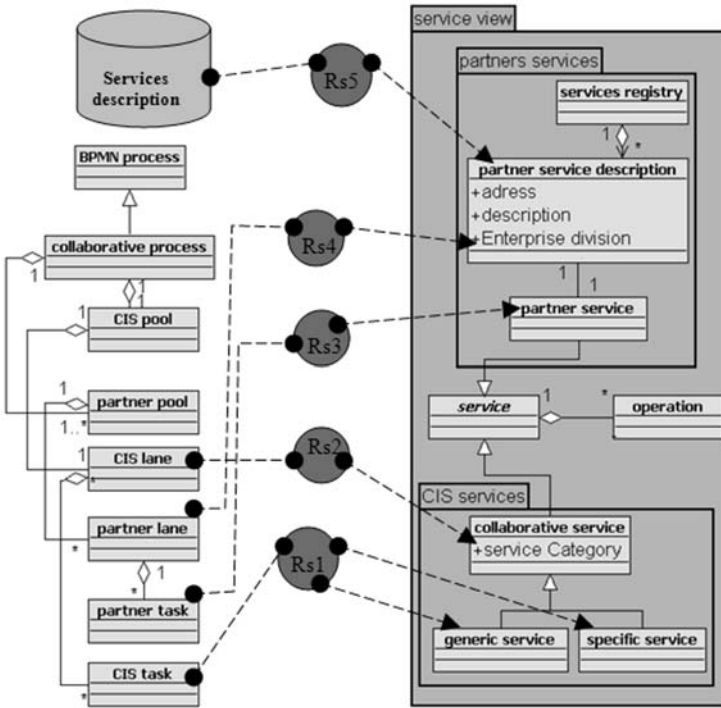


Fig. 4. Localisation of transformation rules for basic generation of the service view

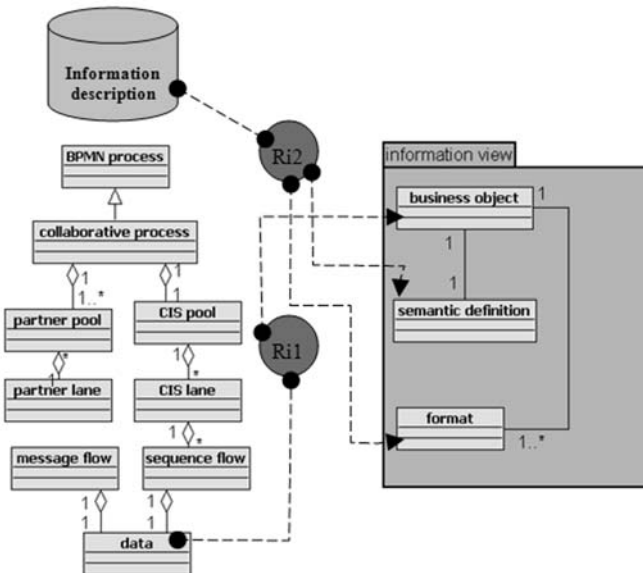


Fig. 5. Localisation of transformation rules for basic generation of the information view



In contrast Fig. 6 is the most developed part of the transformation procedure with nine rules. The “process view” package has been designed using specifications of the BPEL meta model language. BPEL (Business Process Execution Language) is one of the most popular candidates for specification of web services process execution. Some of the rules in Fig. 8 are adaptations of recommendations provided by BPMI when they address the problem of BPMN graph conversion to BPEL well defined XML sentences [8]. It concerns rules *Rp3* to *Rp6*, and rules *Rp8* to *Rp9*. Rules *Rp1*, *Rp2* and *Rp7* participate to the definition of coordination activities.

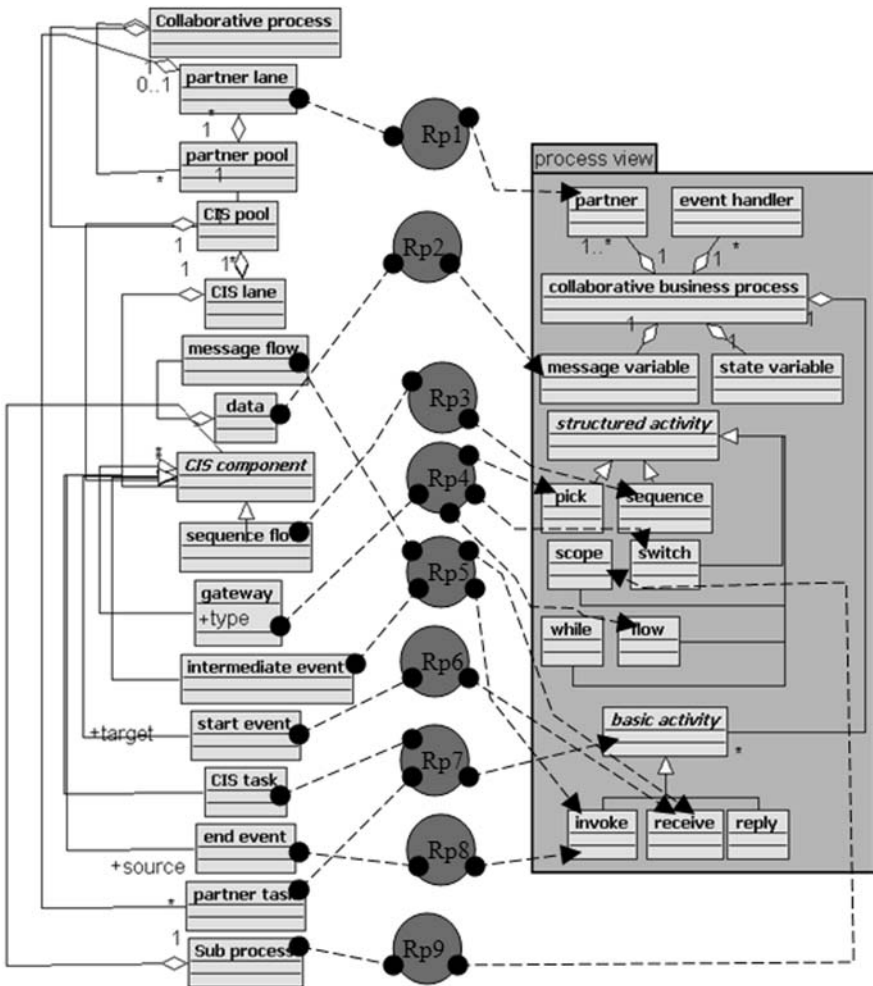


Fig. 6. Localisation of transformation rules for basic generation of the process view

## 4.2 Binding Rules

The binding rules can be used to build the interactions between the elements of the target model coming from the basic generation rules appliance. The links could be inside a target model package or between two different packages (dependence). The goal is to define in the target model the necessary relations that exist in the source model. The relations may be of different types like inheritance, composition, aggregation or simple association. Three binding rules Rb1 to Rb3 are given in the following as an exemple.

**Rb1:** sequence ordering

A *sequence* element issued from Rp3 rule is associated with two basic activities into the same process package.

**Rb2:** information processing

A *service* from service package is related to a *business object* of the information package.

**Rb3:** service identification

A *basic activity* of the process package is linked to a *service* of the service package.

## 5 Prototype Development

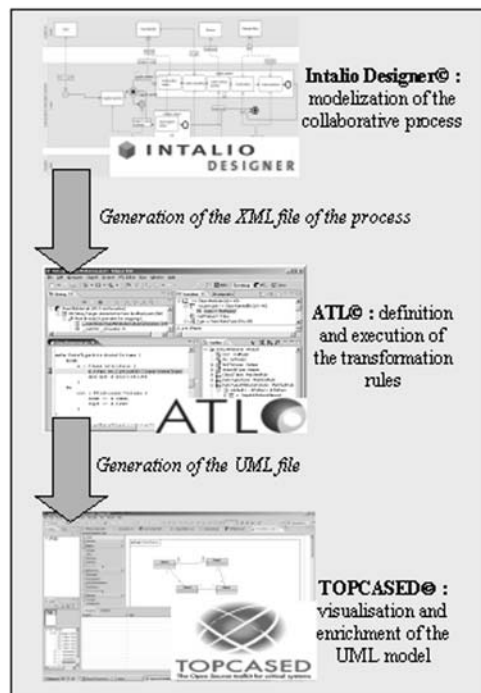


Fig. 7. Technical architecture of the developed prototype

Fig. 7 shows the technical architecture of the prototype, developed to implement our proposition. It is based on three open source tools that run in the **Eclipse**® platform. **Intalio designer**® is a BPM tool that helps user to specify the BPMN model. The **Atlas Transformation Language (ATL)**® can use a process model in the XML format coming from the BPM tool in input, and produce the UML model of the collaborative architecture in output. It is the heart of our transformation prototype. The **TOPCASED**® tool is a computer aided software environment that can perform a graphical edition of the UML model. The ATL tool allows generating a UML model from an XML file that represents the process model. The rules presented previously are coded with ATL. The following gives examples of the ATL code.

### 5.1 Examples of ATL Code

The following ATL code generates the structure of the collaborative architecture (three packages: services, information and processes):

```
rule generatePackages {
  from
  a : BPMN!Collaborativeprocess
  to
  out :UML2!Model
  (
  name<-'Collaborative architecture',
  packagedElement <- OrderedSet {services, information, processes}
  ),
  --generation of services package
  services: UML2!Package (
  name <- 'services view'
  ),
  --generation of information package
  information: UML2!Package (
  name <- 'Information view'
  ),
  --generation of processes package
  processes : UML2!Package (
  name <- 'Process view'
  , packagedElement <- OrderedSet {basic,structured}
  ...
}
```

The following ATL code implements the identified *Rs3* rule (Fig. 4). It permits to generate a partner service (in the collaborative architecture) from a partner task (of the BPMN process).

```
rule generatePartnerServices {
  from
  a : BPMN!PartnerTask
  to
  task :UML2!Class
  (
  name <- a.name
  ),
  service :UML2!Class
  (
  name <- a.name
  )
}
```

The following ATL code shows how to bind the generated ‘partner service’ with the class ‘registry’ of the collaborative architecture. We use the helper (defined function) *CreateAssociation*.

```

for (m in BPMN!PartnerTask.allInstances() ->select(a|true))
{
  --binding registry with partner service
  thisModule.CreateAssociation(registry,thisModule.resolveTemp(m,
  'service'), partners,'appartient a',1,0,'contient','appartient
  à');
}

```

### 5.2 Example of Transformation

A simple example of a collaborative process is proposed in Fig. 8. The collaboration takes place between a customer and a supplier for a trading transaction. Fig. 9 shows the result of the transformation of the collaborative process of Fig. 8 into a collaborative architecture, generated by our prototype. We can see the three packages: service, information and processes and the different links.

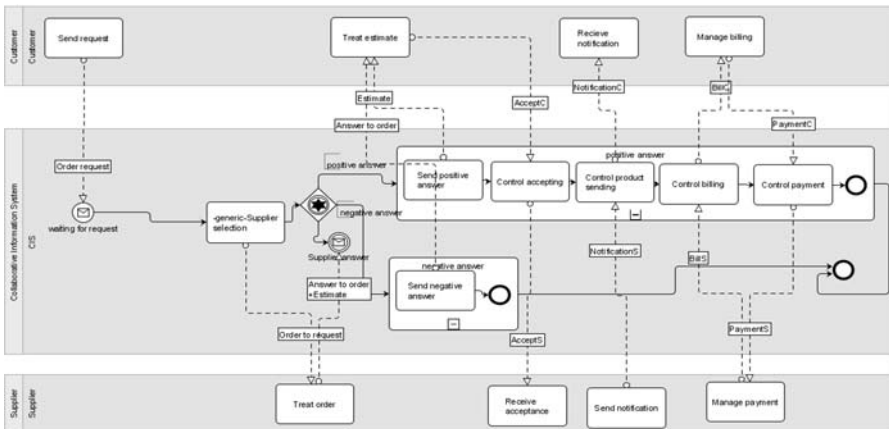


Fig. 8. A BPMN collaborative process model

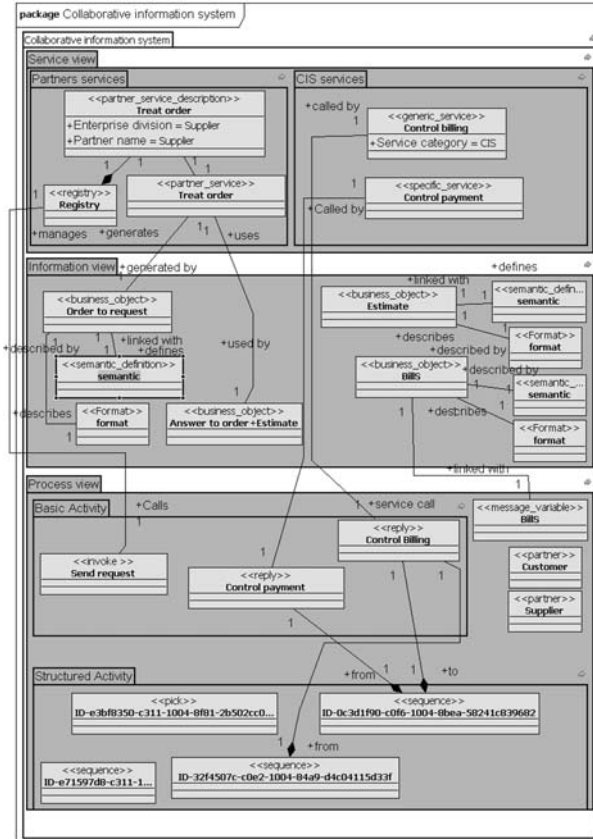


Fig. 9. Result of the transformation using the developed prototype

## 6 Conclusion and Perspectives

Our work is linked to research works in the Model Driven Interoperability field. In this paper we have defined a morphism between collaborative process model (CIM) and collaborative architecture model (PIM). We have showed how we can derive from a CIM model the necessary information to automatically create PIM-level. The generated SOA model can later be mapped to different ICT architectures.

We are aware that it is relatively not frequent to have networks of organizations that are able to draw a collaborative process of their predicted common activities. In [9], we study the contribution of a knowledge based methodology to help in the process model design. The PIM solution that we have produced has been one of the components selected in the JonES project (French project ANR/RNTL 2005). JonES main objective is to test a complete MDA approach in the frame of an

Enterprise Service Bus technology (Target Platform). The solution developed is open source and has been designed by the ObjectWeb community (Petals ESB).

## References

- [1] Grangel Seguer R, Ben Salem, J.P. Bourey, N. Daclin, Y. Ducq.: Transforming GRAI Extended Actigrams into UML Activity Diagrams: a First Step to Model Driven Interoperability, Enterprise interoperability:New challenges and approaches II, Springer edition. ISBN:978-1-84628-857-9, (2007)
- [2] Bourey J.-P. R. Grangel, A.Berre, G. Doumeingts, K. Kalampoukas, M. Bertoni, L. Pondrelli, and N. Daclin: DTG2.1: Report on model establishment, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, n° IST 508-011, (2005)
- [3] Touzi J. : A model transformation for mediation information system design, PhD thesis, Ecole des Mines d'Albi-Carmaux, 9 Nov. (2007)
- [4] Benguria, G., X. Larrucea, B. Elveseater, T. Neple, A. Beardsmore, M. Friess, A Platform Independent Model for Service Oriented Architectures. IESA'06 Conference, Bordeaux, France (2006).
- [5] Bauer B., J.P. Müller, S. Roser: A Decentralized Broker Architecture for Collaborative Business Process Modelling and Enactement, Enterprise Interoperability: New Challenges and Approaches- Springer Verlag - ISBN-10: 1846287138, (2006).
- [6] Darras F., *Proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré*, PhD thesis, Ecole des Mines d'Albi-Carmaux, (2004).
- [7] D'Antonio F.. InterOp Noe Network of Excellence Report, Task Group 2.2 (MoMo), IST 508011, available at <http://www.interop-noe.org>(2005).
- [8] Ouyang, C., W. Van Der Aalst, M. Dumas, A. Hofstede, *Translating BPMN to BPEL*, Technical report - BPM group, Queensland University of Technology, Brisbane (2006).
- [9] Rajsiri V, Lorré JP, Bénaben F and Pingaud H.. Cartography based methodology for collaborative process definition, accepted paper for PRO-VE'07-8<sup>th</sup> IFIP Working Conference on Virtual Enterprises, Portugal (2007).