ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **14**, Number 1, 2011, 67–80

Transportation Process of Containers BPMN-Modeling and Transformation into ACTIF Model

Ismahène HADJ KHALIFA[1], Abdelkader EL KAMEL[1], Pascal YIM[2]

[1]Laboratoire d'Automatique, Génie Informatique et Signal, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d'Ascq Cedex, France
[2]Groupe 3 SI, Cité numérique, 245 Rue Jean Jaurès 59650 Villeneuve d'Ascq Cedex, France
E-mail: `ismahene.hadjkhalifa@centraliens-lille.org`, `abdelkader.elkamel@ec-lille.fr`, `pyim@3suisses.fr`

Abstract. In this paper, we propose to model the transportation process of containers in the Nord-Pas de Calais Region (France). Based on the BPMN (Business Process Modeling Notation) approach, it allows deriving a comprehensive vision of the process and a friendly-user interface of the logistics chain for the analysis and the decision making. Furthermore, ACTIF (Aide à la Conception de systèmes de Transports Interopérables en France), a modeling approach proposed by the French Transportation Ministry, appears to be an interesting alternative approach since it assures the interoperability between the transport systems in France. Hence, we propose a model transformation procedure of the BPMN-model into an ACTIF model.

Key-words: Model Driven Architecture, BPMN model, ACTIF model, Model Transformation.

1. Introduction

Nowadays, the companies evolve in a more and more complex environment: several processes to manage, objectives and constraints to optimize, resources to share.

Furthermore, in order to stick to the fast-evolution environment, the decision-maker has to supervise and control the company performances. To achieve this objective, the business process modeling appears as a key-step to understand the current situation, to identify the dysfunctions and to derive the reasons before any decision- making.

This paper proposes a model of the transportation process of containers in Nord-Pas de Calais Region (France) then a model transformation to assure the interoperability between the transport systems.

The paper is organized as follows. Section 2 introduces a Model Driven Architecture overview. A presentation of the context study is given in Section 3 and of the business process modeling methods state of art in Section 4. In Section 5, we propose a BPMN-model of the transportation process of containers. Finally, in Section 6, we present the technical environment TOPCASED (Toolkit OPen-source for Critical Application & SystEms Development) and the transformation language ATL (Atlas Transformation Language), used to transform the BPMN diagram into an ACTIF model, besides the different steps to achieve this model transformation.

2. MDA Overview

Model Driven Architecture (MDA) is an approach model-use in software developments. It was launched by the Object Management Group (OMG) in 2001. It is a framework for software developments that aims to promote the use of models and their transformations as a fundamental approach for system design. The three primary goals of MDA are portability, interoperability and reusability through architectural separation of concerns [1].

This architecture defines a hierarchy of models from three levels of abstraction: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM) [1], [2].

- ***Computation Independent Model***

A Computation Independent model (CIM) represents the computational independent viewpoint. It is a system model that shows the system in the environment in which it will operate. Hence, it helps in presenting exactly what the system is expected to do.

- ***Platform Independent Model***

A Platform Independent Model (PIM) is a view of the system from a platform independent viewpoint. It focuses on the system operation, hiding the platform specific details, and is independent of the specific technological platform used for implementation. Furthermore, it exhibits platform independence and is suitable for use with a number of different platforms of similar types.

- ***Platform Specific Model***

A Platform Specific Model (PSM) is a view of the system from the platform specific viewpoint. It combines the specifications in the PIM with the details that specify how the system uses a particular type of platform. Hence, it is indispensable for system implementation.

The PIM and PSM can be developed by using any specification language. However, UML (Unified Modeling Language) is typically used since it is a standard modeling language. Furthermore, there are important OMG standards in the core of the MDA [1], [3]: MOF (Meta Object Facility) [4], CWM (Common Warehouse Meta-model) [5], UML [1], XMI (XML Metadata Interchange) [6] and OCL (Object Constraint Language) [7].

The key challenge of MDA is in transforming higher-level models into platform-specific models which can be used to generate the implementation level models. Transforming a model into another model means that a source model is transformed into a target model based on given transformation rules [3].

In this paper, we present a case study, in the transport domain, in which we transformed a BPMN diagram into ACTIF model to guarantee the interoperability in the transport domain.

3. Context study

This study is a part of two French projects carried out by the Nord Pas de Calais Region. The main objective of these projects is to develop interchanges between the multimodal platforms, to blend the procedures and to introduce the automatic interfaces.

This study is part of the first stage. Its main purpose is to introduce an organization which facilitates the harmonization of platform/loader constraints inside Delta 3 platform in Dourges, a town in the Nord-Pas de Calais Region, in order to optimize the local trucking process, i.e. road post-transportation of containers from the fluvial terminal to the warehouses inside the multimodal platform.

To achieve this task, an enterprise process modeling appears as a key-step to understand the current situation, to identify the dysfunctions and hence derive the reasons for better decision-making. First, we studied the different steps to transport the freight from the ship arrival at Dunkerque to the warehouse delivery as well as the interaction between the actors intervening in the transportation supply chain: LDCT (Lille Dourges Conteneur Terminal), the carrier, the loader, the customs service in Dunkerque..., the communications process and the information systems used. This analysis allowed us designing the transportation process of containers.

In the next section, we present the enterprise modeling overview. Then, we select the suitable approach to model the transportation process.

4. Enterprise modeling overview

In literature, several enterprise modeling methods were advocated, among which we quote: GRAI-GIM, UML, ACTIF and BPMN. These methods are briefly presented in this section then the suitable approach is selected to design the transportation process of containers.

4.1. GRAI-GIM method

GRAI-GIM (GRAI Integrated Method) is a method based on enterprise modeling to design or redesign the manufacturing systems. This method focuses mostly on the decision-making level in a company. Its main goal is to improve the global enterprise performances [8]. This method can be applied in several domains such as: manufacturing systems engineering, industrial strategy development, knowledge management. . . The GIM methodology suggests new concepts to describe the decision making systems such as: the GRAI grille and the GRAI network [8]. The GRAI grille is a table. The columns represent the enterprise functions and the lines represent the decision-making levels for the different types of decision such as strategic, tactical and operational decisions with their periods and horizons. The GRAI network is a local representation of the making-decision center. It defines the activities, the initial states and the trigger events of the decision-making process [9].

In conclusion, certainly modeling with the GRAI grille or the GRAI network is clear and easy to understand but it does not present a real interest to our study at this stage.

4.2. UML method

The Unified Modeling Language (UML) is an OMG specification defined by the UML meta-model, a Meta-Object Facility meta-model (MOF). It is a specification language for object modeling. It is a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model. It is designed to specify, visualize, construct, and document the software-intensive systems. UML is not a method since it does not introduce any approach [10]. UML is not restricted to modeling software. It is also used for the business process modeling, the systems engineering modeling and the representing organizational structures.

UML 2.0 [11] has 13 types of diagrams divided into three categories:

– Structure diagram;
– Behavior diagram;
– Interaction diagram.

The most clear and comprehensive diagram to represent the business process is the Activity Diagram since it represents clearly the activities and the flows.

4.3. ACTIF Model

ACTIF stands for “Aide à la Conception de systèmes de Transports Interopérables en France”, a French expression meaning “Design Methodology for Interoperable Transport systems in France”. It is a framework to design the transport system architectures for land transport.

It was developed according to the European ITS (Intelligent Transport Systems) framework architecture and the international standards [12].

The main purpose of the ACTIF framework architecture is to help and guide by means of a method and tools. Hence, it allows having a global vision, identifying and taking into account the multiple actors, exchanging and communicating the information between the actors, having seamless and open systems, assuring the interoperability and the interchangeability of the information between the different systems [12].

The ACTIF model includes the following elements [12]:

- A Functional Area which gathers the functional elements that describe the transport systems activities. The ACTIF architecture introduces nine functional areas.
- A Function which is an information processing operation. The functions originate from the hierarchical breakdown of the functional areas. There are two types of functions: the High-level Function and the Low-level Function.
- A Data Store which represents the elements designed to conserve and provide the information processed in the transport system. They are linked to a functional area and are situated at the different levels within the breakdown of this area.
- A terminator representing the points at which the transport systems interact with their environment. Terminators can be physical/human entities, systems or organizations.
- A Logical Flow which represents the data exchange between the different elements of the model such as: the function, the terminator and the data store. We find the Elementary Flows and the Aggregated Flows which exchange the information between elements in the high level.
- A Standard representing the rules to be used by the specific elements of the model: function, dataflow or terminator.

The ACTIF model represents a reference in the transport domain and a knowledge base comprehensible to the specialists in this domain. Hence, it seems interesting to design the transportation process of containers with ACTIF and we mainly concentrate on the eighth Function Area named “Freight and fleet management” (FA8). However, we note that the ACTIF model does not respect the task synchronization and cannot be implemented.

4.4. BPMN Method

In 2004, the BPMI (Business Process Management Initiative) released the first BPMN specifications version [13]. Later, it joined the OMG (Object Management Group) to create in 2005 a Task Force, focused on BMI (Business Modeling & Integration), called BPMN (Business Process Modeling Notation). The primary goal is to provide a notation readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the manager who will monitor those processes.

BPMN is a notification which represents a business process disjointing the business information and the technical information. A BPMN diagram is similar to UML Activity Diagram however the BPMN asset is to create a standardized bridge between the business process design and the process implementation [14]. Hence, following the business process modeling, we can generate automatically the BPEL process (Business Process Execution Language) to implement with the process engine [15].

However, the process execution aspect will not be developed in this paper. Since our main objective here is to design containers transportation process, the execution aspect will be presented in a future work.

The latest BPMN specifications version was released in 2006. It defines four basic elements in a business process diagram (White, 2007):

- The Flow Objects which are the main graphical elements to define the behavior of a Business Process. There are three flow objects: the activity, the event and the gateway.
- The Connecting Objects which represent the different ways of connecting the Flow Objects to each other or other information.
- The Swimlanes which gather and organize the activities to differentiate different responsibilities of participants. There are two types of swimlanes: The Pool which represents a participant or a business entity in a process and the Lane which is a sub-partition within a Pool.
- The artifacts which are used to provide additional information about the Process.

In conclusion, we choose to design the transportation process of containers using the BPMN method because the BPMN notation is an OMG standard which produces clear diagrams and assures task synchronization. Moreover, it contains different elements which describe the business process and the messaging between the processes and can be transformed in a future step into an executable language (BPEL). However, it will be interesting to design our process using the ACTIF model since it is used in transport domain in France and assures the interoperability between the systems in this domain. That is why; we propose to transform the BPMN diagram into an ACTIF model in the last section.

5. Modeling transportation process of containers with BPMN

In this section, we design the transportation process of containers from Dunkerque port to the loader. We use "Intalio Designer" which is an open source tool to design the business processes [16].

The actors are represented by Pools with their activities. The process begins with an empty circle representing a start event. Then, we represent the activities of each actor in the corresponding pool. Moreover, we use intermediate events such as:

– The “Message” represented inside the “Pool LDCT”. It means that this actor must wait until he receives an e-mail from the NCS actor to execute the activity “Unloading of the barge”.

– The “Timing” is used in the same pool. It means that LDCT must wait 5 to 30 minutes to go to the “Pile maritime”. Figure 1 illustrates this.

Furthermore, inside the “Pool LDCT”, there is a gateway used to represent two activities setting in the same time.

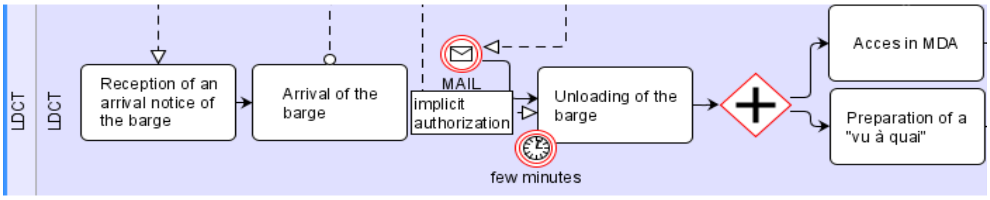

Fig. 1. Use of intermediate events and gateway.

We use Sequence Flow to connect two Flow Objects if they are inside the same Pool. Else, we use Message Flow. Finally, we finish the diagram with an end event. Figure 2 illustrates the local trucking process inside Delta 3 platform.

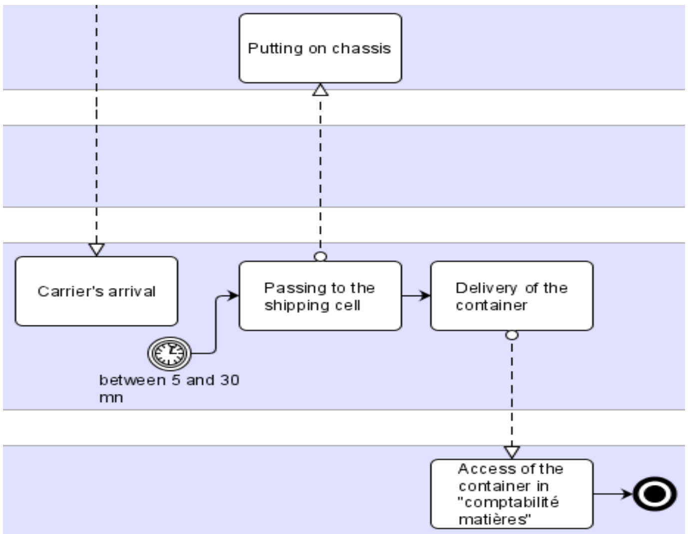

Fig. 2. Local trucking process.

We propose, in the following section, to transform the BPMN diagram into an ACTIF model in order to take advantage of both modeling methods, mainly interoperability of transport systems besides the BPMN key-advantages.

6. Transforming BPMN diagram into ACTIF model

6.1. Technical environment description

We choose TOPCASED environment to create the source and target meta-models, to generate the model corresponding to the source meta-model and to transform the models. It is an open source environment based on Eclipse. We download TOPCASED 1.0.0. version available in the website `http://gforge.enseeiht.fr/frs/?group_id=7`.

It is an integrated System/Software engineering toolkit compliant with the requirements of critical and embedded applications. It is a result of TOPCASED project which was made to encourage tools cooperation. Its main goals are to perpetuate methods and tools for system and software development, to ensure the independence of development platforms, to be able to adapt the tools to the process, and to take into account qualification constraints [17].

TOPCASED is strongly model-oriented. So, it provides the model editors, the model checkers and the model transformations and it is itself based on modeling and code generation. Furthermore, we can develop our own graphical editors and model transformations. To do that, TOPCASED uses tools based on MDA, MDE (Model Driven Engineering) and QVT (Queries Views Transformations) norms [18].

In TOPCASEE project, several tools open source were tested such as MTF, ATL and MTL. However, ATL seems to be the most interesting since it is complying with OMG/MDA/MOF/QVT and we can use it with an open source tool. Furthermore, it respects the OMG standards since it is built on top of OCL. In consequence, ATL was chosen to transform models in TOPCASED environment [18].

ATL is a transformation language proposed and developed by ATLAS INRIA & LINA. It is hybrid since it contains a mixture of declarative and imperative constructs. Furthermore, ATL transformations are unidirectional, operating on read-only source models and producing a write-only target model [19].

An ATL transformation contains three parts [19], [20]:

- **The header**

It is used to declare general information such as the module name, the input and output meta-model names.

- **The helpers**

The helpers are functions based on OCL standard. They have a name, a context and a type. OCL defines two kinds of helpers: operation helpers and attribute helpers.

- **The rules**

The rules are the core of ATL transformations because they describe how elements of a source model should be translated into elements of a target model. So, they express a mapping between an input element and an output element. They may be specified either in an imperative style or in a declarative style.

Several transformations were realized using ATL such as "EMF to KM3", "Grafcet to PetriNet", "MOF to UML", "UML to Java". These transformations are available on the website `http://www.eclipse.org/m2m/atl/atlTransformations/`.

Furthermore, a recent article [21] presents a transformation GRAI into UML Activity Diagram. For this, Grangel and Al. present the mapping between source model elements and target model elements. After that, they define the helpers such as a helper to test whether a GRAI activity has subactivities or not. So, a GRAI activity is transformed into a UML Action if it does not have subactivities, otherwise it is transformed into a UML CallBehaviorAction and its called Activity.

6.2 Transformation of the BPMN model into an ACTIF model

To transform the BPMN model into an ACTIF model, we must follow these steps:
Step 1: Creation of the source and target meta-models
Step 2: Creation of the transformation rules
Step 3: Creation of the source model
Step 4: Configuration
Step 5: Result

Step 1: **Creation of the source and target meta-models.** We created the meta-models in ECORE format using TOPCASED ECORE diagrams.

◦ *Creation of the source meta-model.* We remember that a BPMN diagram contains the Flow Objects, the Connecting Objects, the Swimlanes and the Artifacts. A Flow Object can be an Activity, a Gateway or an Event. There are three kinds of Events: Start, Intermediate or End Event. There are three types of Connecting Objects: the Sequence Flow, the Message Flow and the Association. Pools and Lane are two kinds of Swimlane. We use the Sequence Flow to connect two Flow Objects if they are inside the same Pool. Else, we use the Message Flow. An Association links a Flow Object to an Artifact. These elements form the BPMN meta-model. It is illustrated by Figure 3.

◦ *Creation of the target meta-model.* To create the ACTIF meta-model, several discussions were made with the chief of the ACTIF project to define the good meta-model. The ACTIF model contains nine Functional Areas, high-level functions and low-level functions, Data Stores, Logical Flow and Terminators. The meta-model ACTIF is illustrated by Figure 4.

Step 2: **Creation of the transformation rules.** Transformation rules describe how a Source Model is transformed into a Target Model. Table 1 describes the mapping between them.

We didn't find the corresponding element of the Data Store in the BPMN model. It can be considered like a Low-level Function but it is designed to conserve and provide information processed in the transport system. Furthermore, the Standards which represent the rules to be used by the specific elements of the ACTIF model haven't any corresponding element in BPMN model. So, when we transform the BPMN model into an ACTIF model, these elements will be missed.

Step 3: **Creation of the source model.** To create the source model, we use the EMF (Eclipse Modeling Framework) to define and execute the data model. It is an

Eclipse-based modeling framework and code generation facility for building tools and other applications based on a structured data model [22], [23].

In our case, we decided to design a part of containers transportation process. It is the local trucking part. So, we get the model illustrated in Figure 5.

Step 4: **Configuration.** After the creation of the source model, the configuration enables to get the target model. So, in our case, two elements in ACTIF model don't have any corresponding element in the BPMN model. Consequently, when the BPMN model is transformed into an ACTIF model, these elements will be missed and then we will lose some information conserved in the Store Data of the transport system and the rules to be used by the specific elements of the ACTIF model.

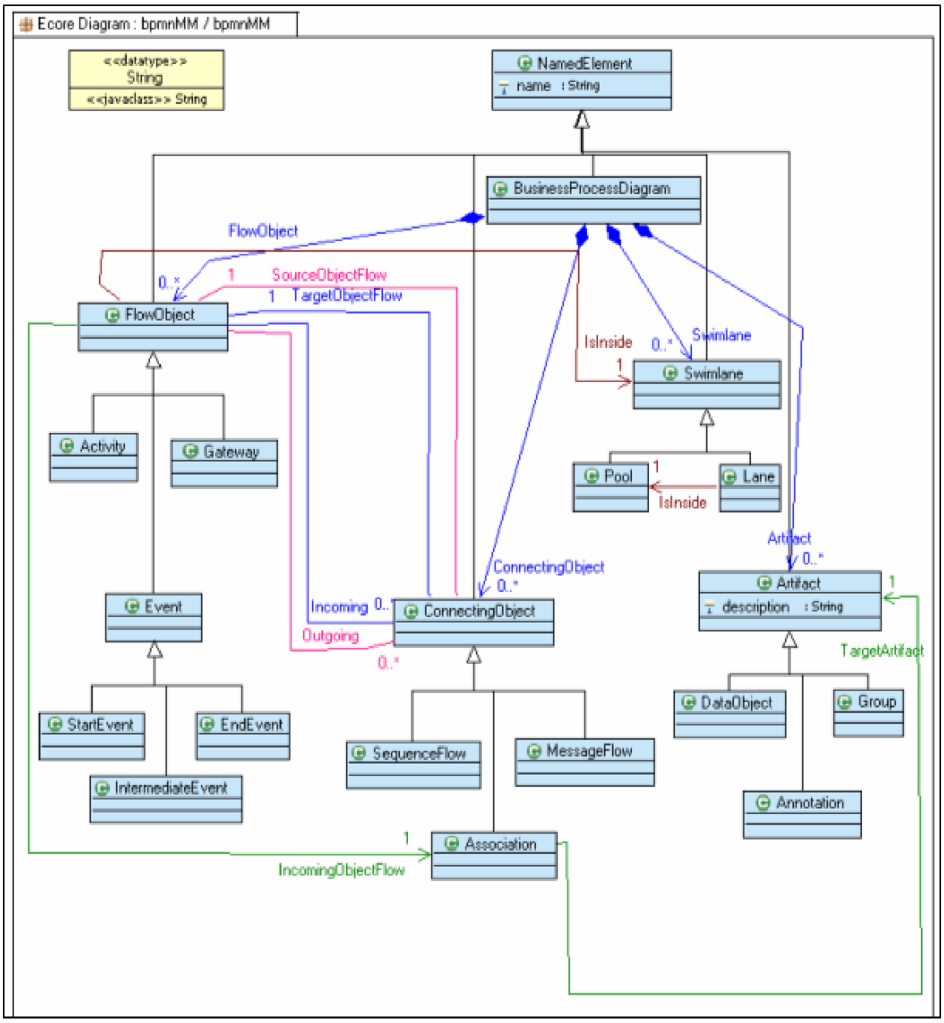

Fig. 3. BPMN Meta-model.

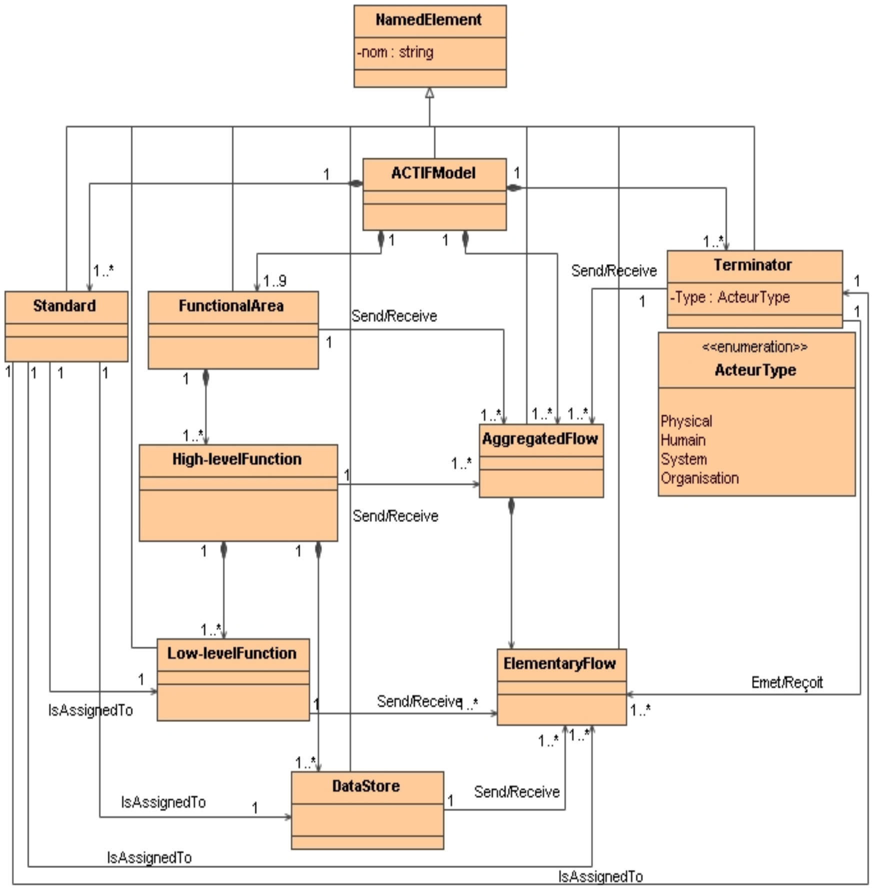

Fig. 4. ACTIF Meta-model.

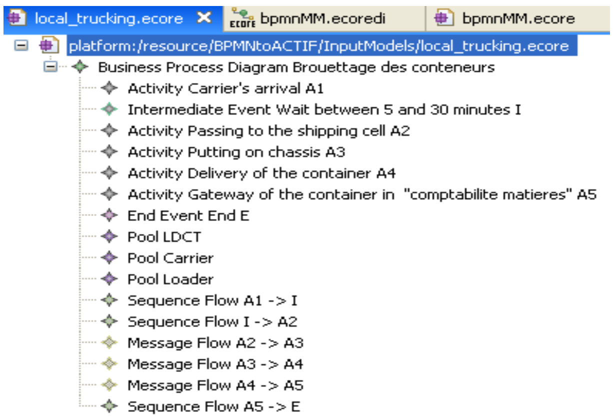

Fig. 5. Source model.

Table 1. Simple Mapping between BPMN Diagram and ACTIF Model

Source (BPMN)	Target (ACTIF)	Type of mapping	Observations
Pool	Terminator or Functional Area	One to many	A Pool groups and organizes the activities to differentiate the different responsibilities of the participants in a process. If it represents a participant then it will be mapped into a Terminator. While, if it represents a business entity then it will be transformed into a Functional Area.
Lane	High-level Function	One to one	A Lane is a sub-partition within a Pool. It is used to organize and categorize activities. It corresponds exactly to the definition of the High-level Function.
Flow Objects	Low-level Function or Data Store	One to many	A Flow Object defines the behavior of a Business Process then it can be mapped to a Low-level Function.
?	Data Stores	–	–
Message Flow	Aggregated Flow	One to one	A Message flow is used to show the flow of messages between two pools then it can be mapped into an Aggregated Flow if it links two elements in the high level.
Sequence Flow	Elementary Flow	One to one	A Sequence Flow links Flow Objects inside the same Pool then it will be transformed into an Elementary Flow.
?	Standard	–	–

7. Conclusion

In this paper, we proposed a model of the transportation process of containers in Nord-Pas de Calais Region. We selected the BPMN method because the BPMN notation is an OMG standard which produces clear diagrams and assures task synchronization. Moreover, it includes the different elements which describe the business process and the messaging between the processes and can be transformed into an executable language. However, it is interesting to design our process with ACTIF model since it is used in transport domain in France and assures the interoperability between systems in this domain. For this reason, we transformed the BPMN diagram into an ACTIF model. To carry out this task, we used TOPCASED as a transformation environment and ATL as a transformation language. After creating the source meta-model and the target meta-model with TOPCASED Diagram, we proposed a mapping between them. However, transforming models is not evident since in our case, there is any corresponding element in the source meta-model for Standards and Data Stores. Consequently, we risk losing information.

Hence, it will be interesting to concentrate on this, in the coming researches, because designing using the ACTIF model allows the company to interoperate with the other actors in the transport domain and designing using BPMN, an OMG standard, allows the enterprise to understand better the current situation before making any decision.

Acknowledgments. We wish to thank M. Alain Demeunynck, the General-Secretary of Club Logistique & Transport 59/62, M. Jean Pierre Grassien, the Project Chief, Mrs. Nelly Severac, the Exploitation Manager of LDCT, and M. Yannick Denis, the ACTIF Project Chief, for their constructive advices.

References

[1] *OMG: MDA Guide Version 1.0.1*, Document number: omg/2003-06-01 edition, `http://www.omg.org/docs/omg/03-06-01.pdf`, 2001.

[2] Bézivin, J., Gérard S., Muller P.A., Rioux L., *MDA components: Challenges and Opportunities*, Workshop on Metamodelling for MDA, pp. 23–41, 2003.

[3] Kadima H., *MDA, Conception orientée objet guidée par les modèles*, Dunod, Paris, 2005.

[4] *OMG, Meta Object Facility (MOF) Core Specification, version 2.0*, OMG document formal/06-01-01, `http://www.omg.org/docs/formal/06-01-01.pdf`

[5] *OMG: Common Warehouse Metamodel (CWM) Specification, version 1.1*, volume 1. OMG document formal/03-03-02, `http://www.omg.org/docs/formal/03-03-02.pdf`, 2002.

[6] *OMG: XML Metadata Interchange (XMI) specification, version 2.0*, OMG Document formal/2003-05-02, `http://www.omg.org/cgi-bin/doc?formal/2003-05-02`, 2002.

[7] *OMG: OCL 2.0 Specification, version 2.0*, OMG document ptc/2005-06-06, `http://www.omg.org/docs/ptc/05-06-06.pdf`, 2006.

[8] Doumeingts G., Vallespir V., Zanettin M., Chen D., *GRAI Integrated Methodology: A Methodology for Designing CIM Systems. Version 1.0*, GRAI/LAP, University Bordeaux I, 1992.

[9] Pierreval H., *Les méthodes d'analyse et de conception des systèmes de production*, Edition Hermès, 1990.

[10] Rumbaugh J., Jacobson I., Booch G., *UML 2.0 Guide de référence*, Edition CampusPress, deuxième édition, 2005.

[11] *OMG: Unified Modeling Language: Superstructure. Version 2.0*, formal/05-07-04, 2004.

[12] *ACTIF, website of the project ACTIF*, `http://www.its-actif.org`

[13] *BPMN: Business Process Modelling Notation Specification*, May 2004.

[14] *BPMN: Business Process Modelling Notation Specification*, February 2006.

[15] White A. S., *Using BPMN to Model a BPEL Process*, IBM Corporation, 2007.

[16] Helkiö P., Seppälä A., Syd O., *Evaluation of Intalio BPM Tool*, Special Course in Information System Integration, 2006.

[17] Farail P., Gaufillet P., *TOPCASED un environnement de développement Open Source pour les systèmes embarqués*, article published by TOPCASED consortium members, `http://www.topcased.org/`, 2005.

[18] Canals A., Le-Camus C., Feau M., Jolly G., Bonnafous V., Bazavan P., *Une utilisation opérationnelle d'ATL : l'intégration de la transformation de modèles dans le projet TOPCASED*, article published by the TOPCASED consortium members, `http://www.topcased.org/`, 2005.

[19] Jouault F., Kurtev I., *Transforming Models with ATL*, *Proceedings of the Model Transformations in Practice Workshop*, *MoDELS 2005*, Montego Bay, Jamaica, 2005.

[20] Allilaire F., Bézivin J., Jouault F., Kurtev I., *ATL: Eclipse Support for Model Transformation*, *Proceedings of the Eclipse Technology eXchange Workshop (eTX) at the ECOOP 06 Conference*, Nantes, France, 2006.

[21] Grangel R., Ben Salem R., Bourey J. P., Daclin N., Ducq Y., *Transforming GRAI Extended Actigrams into UML Activity Diagrams: a First Step to Model Driven Interoperability*, *3rd International Conference, Interoperability for Enterprise Software and Applications*, *I-ESA 07*, Portugal, March 2007.

[22] Parrend P., *Introduction pratique au Développement orienté Modèle*, March 2005.

[23] Catherine G., *Using EMF*, IBM, May 2003.