Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models

Alfonso Rodríguez[1], Eduardo Fernández-Medina[2], and Mario Piattini[2]

[1] Computer Science and Information Technology Department
University of Bio-Bio,
Chillán, Chile
alfonso@ubiobio.cl
[2] ALARCOS Research Group, Information Systems and Technologies Department,
UCLM-Indra Research and Development Institute,
University of Castilla-La Mancha
Ciudad Real, Spain
{Eduardo.FdezMedina,Mario.Piattini}@uclm.es

Abstract. Nowadays, business process modeling, using industrial standards such as UML or BPMN, offers us a good opportunity to incorporate requirements at high levels of abstraction. In the context of Model Driven Architecture (MDA), the business process model is considered as a Computation Independent Model (CIM). In our proposal we will transform the business process specifications into analysis-level classes and use cases, both of which are UML artifacts used to describe the problem in the context of Platform Independent Models (PIM). Such artifacts are complementary, as they are only a subset of the analysis-level classes and use cases that describe the whole problem, in the first stages of the software development process. This work contains the principle issues involved in the main standards that allow us to represent a business process, details of the transformation rules in QVT specification and an illustrative example in which our proposal has been applied.

1 Introduction

A Business Process (BP) is the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result. A model is a simplified view of a complex reality. It is a means of creating abstraction, thus allowing one to eliminate irrelevant details and focus on one or more important aspects at a time. Business process models enable a common understanding and facilitate discussion among different stakeholders in the business [1, 9].

Furthermore, requirement specification usually results in a specification of the software system which should be as exact as possible [2], since effective business process models facilitate discussion between different stakeholders in the business, allowing them to agree on the key fundamentals and to work towards common goals [9].

Several languages and notations exist for business process modeling [10] and of these, the Unified Modeling Language (UML) and the Business Process Modeling Notation (BPMN) are widely accepted standard notations [15].

I.-Y. Song et al. (Eds.): ER Workshops 2008, LNCS 5232, pp. 103–112, 2008.

The Model Driven Architecture (MDA) approach [16] is not based on one single idea. Among its objectives are: the separation of business-neutral descriptions and platform dependent implementations, the expression of specific aspects of a system under development with specialized domain-specific languages, the establishment of precise relationships between these different languages in a global framework and, in particular, the capability of expressing operational transformations between them [4]. For the transformation of models, OMG has proposed Query/View/Transformation (QVT) [18], in order to seek an answer which is compatible with its MDA standard suite: UML, MOF, OCL, etc [12].

Our proposal is within the MDA scope, and we have used the QVT transformation to move from a business process model (CIM) to analysis-level classes and use cases (PIM). The artifacts obtained can complement the requirements captured in a software development process. For this purpose, we have chosen the UP (Unified Process) [11], which is composed of a set of activities necessary for transforming users' requirements into a software system, due to the fact that it is a consolidated and successful software construction method.

The structure of the rest of the paper is as follows: in Section 2, we shall summarize the main issues in relation to business processes modeling. In Section 3, we shall present CIM to PIM transformations, for analysis-level classes as well as use cases. Finally, in Section 4, we shall present an illustrative example and in Section 5 our conclusions will be drawn.

2 Business Processes Modeling

In business process modeling, the main objective is to produce a description of a reality (for example, the way in which a commercial transaction is carried out) to understand and eventually modify it with the aim of incorporating improvements into it. As a consequence, it is important to have a notation that allows us to model the essence of the business as clearly as possible. This notation must allow us to incorporate different perspectives which give place to different diagrams in which the rules, goals and objectives of the business, and not only relationships but also interactions, are shown [7]. A significant part of the success of modeling has to do with its ability to express the different needs of the business as well as its having a notation in which these needs can be described. This is why, when choosing an approach and/or notation, the properties of the object to be modelled must be taken into account, in other words, the business process, the environmental features and the underlying reasons for its use [5].

Among those techniques that have been used for business process modeling are the following: flow diagrams, data flow diagrams, entity-relationship diagrams, state-transition diagrams, Gantt charts, Role Activity Diagrams (RAD), the family of techniques known as Integration Definition for Function Modeling (IDEF), Petri Nets, simulation, techniques based on knowledge (artificial intelligence) and workflow techniques [1, 10]. At present, and according to the state of the business process modeling industry [15], it is possible to identify UML [17] and BPMN [6] among the main standards.

In UML 2.0 the element used to represented business process and workflows is the Activity Diagrams (UML 2.0-AD) [13]. In previous UML versions, expressivity was limited and this fact confused users who did not use orientation towards objects as an approach for modeling. However, it is now possible to support flow modeling across a wide variety of domains. The UML 2.0-AD elements that will be used in our proposal are Activity Partition, Action, Data Store Node, and Interruptible Activity Region.

In BPMN the elements used to process representation is the Business Process Diagram (BPMN-BPD). This diagram was designed to facilitate its use and understanding, and to offer an expressive force with which to model complex businesses. The BPMN-BPD elements that will be used in our proposal are Pool, Lane, Data Objects, Group, and Activity.

3 CIM2PIM Mappings

In our opinion, a business process which has been built by a business analyst is not only useful in the specific business field, but is also very useful in a software construction process. From this process we can obtain system requirements, a stage taken into account by all modern development processes that basically consists of obtaining from the customer or the interested parties the system requirements for developing software construction from this point. In our proposal, CIM2PIM transformations are aimed at obtaining useful artifacts in software development. Both the analysis-level classes and the use cases obtained from the business process model become part of an ordered and systematic process of software development.

A fundamental aspect MDA models transformation. The Object Management Group (OMG) proposal which allows us to perform this task is Query/View/ Transformation (QVT) [18]. QVT is compatible with the MDA standard since its abstract syntax is defined as a MOF 2.0 (Meta Object Facility) metamodel. Basically, QVT offers us the possibility of manipulating models by considering queries that use a model as an input. It selects specific elements of this model according to a search pattern, views corresponding to models that are derived from other models, and finally transformations, which use one or more input models to obtain an output model or result as a reference.

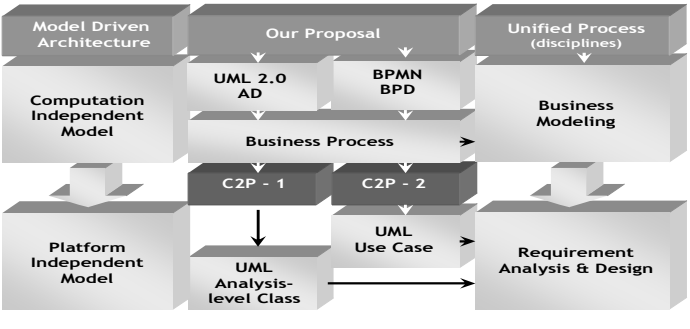

Fig. 1. Our proposal overview

In Figure 1, the basic aspects of our proposal are shown. The BP model can be specified with UML 2.0-AD or BPMN-BPD. In an MDA approach, such a description corresponds to a computation independent model. Through the application of a set of transformation rules described with QVT (C2P-1 y C2P-2) it is possible to obtain a subset of the analysis-level classes and use cases that facilitate the understanding of the problem. UP is considered in our proposal because both the BP description and the artifacts obtained through the transformation can be used during the first disciplines. Thus, the business description will be useful in the "Business Modeling" discipline and both analysis-level classes and use cases complement the "Requirement" and "Analysis & Design" disciplines.

The set of transformation rules that we propose consider as an input a BP model described with UML 2.0-AD. In order to include models described with BPMN-BPD, we have established an equivalence relationship between the elements of both notations. Although the relationship between these two notations is dealt with in [22], the equivalents that we have proposed permit a relationship to exist between the concepts which represent similar business process concepts. These equivalence relationships have been expressed using QVT language (see Table 1).

Table 1. Mapping between BPMN-BPD and UML 2.0-AD elements

```
transformation BPMN-BPD2UML-AD
  top relation R1   // from Pool to Activity Partition
  {
    checkonly domain bpmn_BusinessProcessDiagram p:Pool {name = n}
    enforce domain uml_ActivityDiagram ap:ActivityPartition {name = n}
  }
  top relation R2   // from Lane to Activity Partition
  {
    checkonly domain bpmn_BusinessProcessDiagram l:Lane {name = n}
    enforce domain uml_ActivityDiagram ap:ActivityPartition {name = n}
  }
  top relation R3   // from Group to Interruptible Activity Region
  {
    checkonly domain bpmn_BusinessProcessDiagram g:Group {name = n}
    enforce domain uml_ActivityDiagram ir:InterruptibleActivityRegion {name = n}
  }
  top relation R4   // from Activity to Action
  {
    checkonly domain bpmn_BusinessProcessDiagram ac:Activity {name = n}
    enforce domain uml_ActivityDiagram act:Action {name = n}
  }
  top relation R5   // from Data Object to Data Store Node
  {
    checkonly domain bpmn_BusinessProcessDiagram do:DataObject {name = n}
    enforce domain uml_ActivityDiagram dsn:DataStoreNode {name = n}
  }
  top relationR6   // from Start Event to Initial Node
  {
    checkonly domain bpmn_BusinessProcessDiagram s:StarEvent {name = n}
    enforce domain uml_ActivityDiagram act:Action {name = n}
  }
```

In sections 3.1 and 3.2, we have given a detail description of the transformations from a Business Process to analysis-level classes and use cases respectively. In each case, the set of rules and their specification in QVT text form will be shown.

3.1 Business Process to Analysis-Level Class (C2P-1)

In this section, we will present the set of rules that allows us to obtain analysis-level classes from a business process specification.

In our review of the literature related to this subject, only two works dealing directly with this type of transformations were found. In the first [3], activity diagrams are transformed into analysis classes. This transformation is not automatically performed and a previous version of UML 2.0 is used. In the second work [20], the software designer studies the BP model described with BPMN by extracting the UML classes which are later refined. The difference between these proposals and ours is that, in both cases, the generation process of analysis classes is manually performed and the result of transformations is not related to a software development process.

The transformation from a business process model specified with UML 2.0-AD (or its equivalence in BPMN-BPD) to analysis-level classes are described with QVT language in Table 2.

Table 2. Mapping between Activity Diagrams and Class Diagrams elements

```
transformation ActivityDiagram2ClassDiagram
  top relation R1   // from Activity Partition to Class
  {
    checkonly domain uml_ActivityDiagram ap:ActivityPartition {name = n}
    enforce domain uml_ClassDiagram c:Class {name = n}
    where {
         ap.containedNode → forAll(cn:Action|R4(cn))
    }
  }
  top relation R2   // from Interruptible Activity Region to Class
  {
    checkonly domain uml_ActivityDiagram iar:InterruptibleActivityRegion {name = n}
    enforce domain uml_ClassDiagram c:Class {name = n}
  }
  top relation R3   // from Data Store Node to Class
  {
    checkonly domain uml_ActivityDiagram dsn:DataStoreNode {name = n}
    enforce domain uml_ClassDiagram c:Class {name = n}
  }
  relation R4   // from Action to Operation in Class
  {
    checkonly domain uml_ActivityDiagram ac:Action {name = n, inPartition=ap}
    enforce domain uml_ClassDiagram op:Operation {name = n, ownerClass=c:Class{name=ap.name}}
  }
```

Additionally, we present the set of rules that permits analysis-level class refinement (see Table 3). These rules are applied later than QVT rules. Their main objective is that of enriching the class model through the incorporation of meaningful region names, the identification of relationships between the classes obtained and the elimination of redundant elements.

Table 3. Refinement Rules for Analysis-Level Classes

RR 1:	Region Name is obtained by linking the ActivityPartition names where the InterruptibleActivityRegion is contained
RR 2:	Composition relationships are obtained from top and middle ActivityPartitions
RR 3:	Redundant specifications must be eliminated

3.2 Business Process to Use Case (C2P-2)

In this section, we will present the set of rules that permit us to obtain use cases from a business process description.

In our review of literature, we discovered that in [19], the possibility of manually obtaining use cases from a BP specification made with BPMN is suggested. In [14], the automatic attainment of UML artifacts from a BP description that was made using BPMNN is proposed. The authors extend the BPMN (Extension Level 1) in order to add information about the sequence and the input and output flows. This allows them to apply rules from which use cases, state diagrams, sequence and collaboration are attained. In [21], a manually performed transformation from a BP described with AD-UML 2.0 to use cases is stated and finally, in [8], use cases are obtained from business process models which are not represented by activity diagrams. The differences between these proposals and ours are basically the following: (i) even in works where there are automatic transformations, previous manual intervention is required, ii) transformations are not described by using languages which have been specially designed for this purpose and iii) the result of the transformations does not appear to be linked to a business process development.

The transformations from a business process model specified with AD-UML 2.0 (or its equivalence in BPD-BPMN) to use cases are carried out in accordance with the QVT rules described in Table 4.

Table 4. Mapping between Activity Diagrams and Use Case elements

```
transformation ActivityDiagram2UseCaseDiagram
  top relation R1   // from Activity Partition to Actor
  {
    checkonly domain uml_ActivityDiagram ap:ActivityPartition {name = n}
    enforce domain uml_UseCaseDiagram a:Actor{name = n}
    where {
          ap.containedNode → forAll(cn:Action|R3(cn))
    }
  }
  top relation R2   // from Interruptible Activity Region to Actor
  {
    checkonly domain uml_ActivityDiagram iar:InterruptibleActivityRegion {name = n}
    enforce domain uml_UseCaseDiagram a:Actor {name = n}
    where {
          ap.containedNode → forAll(cn:Action|R3(cn))
    }
  }
  relation R3   // from Action to UseCase
  {
    checkonly domain uml_ActivityDiagram ac:Action {name = n, inPartition=ap}
    enforce domain uml_UseCaseDiagram uc:UseCase {name = n, subject= ACTORS: Set(Actor)};
    where {
          ACTORS→including (a:Actor{name=ap.name})
    }
  }
```

Table 5. Refinement rules for Use Cases

RR 1:	Subject name is obtained from the business process name
RR 2:	Region Name is obtained by linking the Activity Partition names where Interruptible Activity Region is contained
RR 4:	Main Actor corresponds to the Activity Partition or region name where Initial Node is present
RR 5:	Actor Generalization is obtained from top and middle Activity Partitions
RR 6:	Redundant specifications must be eliminated

We also present a set of rules that permits use case refining (see Table 5). These rules are applied later than QVT rules. Their main objective is that of enriching the use case through the incorporation of the subject name, region names, identifying the main actor, actor generalization and the elimination of redundant elements.

4 Example

Our illustrative example (see Figure 2) describes a typical business process for the admission of patients to a health-care institution. In this case, the business analyst identified the following Activity Partitions: Patient, Administration Area (which is a top partition that is divided into the Admission and Accounting middle partitions), and the Medical Area (divided into Medical Evaluation and Examinations). We shall apply the transformations described in the previous section to this business process.

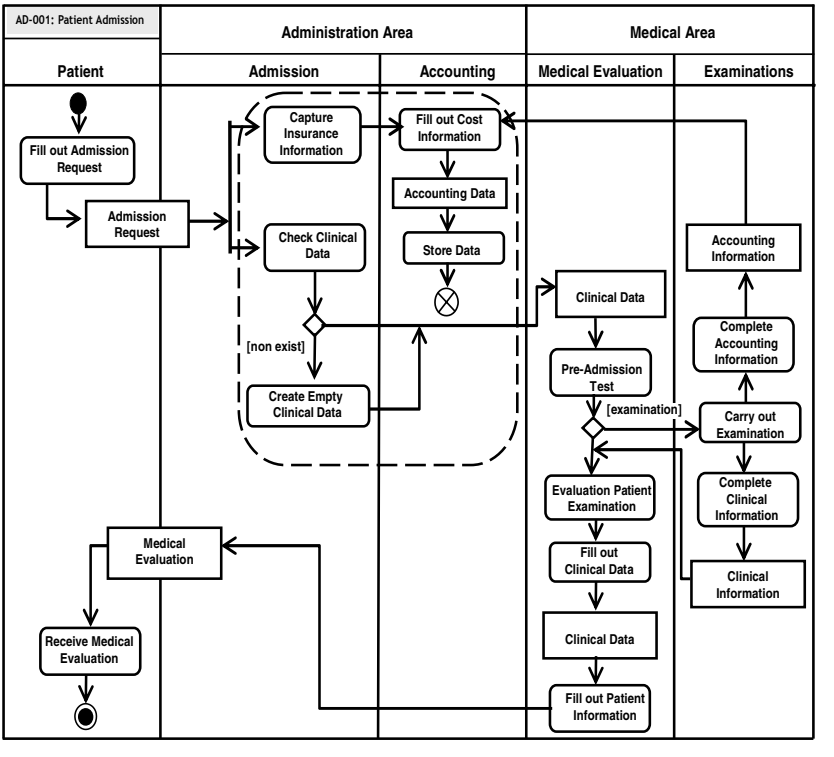

Fig. 2. Admission of Patients to a Medical Institution

The result of the application of the QVT and refinement rules for obtaining analysis-level classes and use case elements from the example is shown in Table 6.

Table 6. Mapping from UML 2.0-AD to analysis-level classes and use cases elements

Rule	UML2.0-AD elements	To	Analysis-level class element
C2P-1/R1	Activity Partition	Class	Patient, Administration Area, Admission, Accounting, Medical Area, Medical Evaluation, Examinations
C2P-1/R2	Interruptible Activity Partition	Class	Region 01 (AdmissionAccounting)
C2P-1/R3	Data Store Node	Class	Admission Request, Accounting Data, Clinical Data, Accounting Information, Clinical Information and Medical Evaluation
C2P-1/R4	Action	Operation	Fill out Admission Request, Receive Medical Evaluation, Capture Insurance Information, Check Clinical Data, Create Empty Clinical Data, Fill out Cost Information, Store Data, Pre-Admission Test, Evaluate Patient Examinations, Fill out Clinical Data, Fill out Patient Information, Complete Accounting Information, Carry out Examinations and Complete Clinical information
C2P-2/R1	Activity Partition	Actor	Patient, Administration Area, Admission, Accounting, Medical Area and Medical Evaluation and Examinations
C2P-2/R2	Interruptible Activity Region	Actor	Region 01 (AdmissionAccounting)
C2P-2/R3	Action	Use Case	Fill out Admission Request, Receive Medical Evaluation, Capture Insurance Information, Check Clinical Data, Create Empty Clinical Data, Fill out Cost Information, Store Data, Pre-Admission Test, Evaluate Patient Examinations, Fill out Clinical Data, Fill out Patient Information, Complete Accounting Information, Carry out Examinations and Complete Clinical information

In Figure 3, analysis-level classes obtained from a business process specification for patient admission are graphically shown.

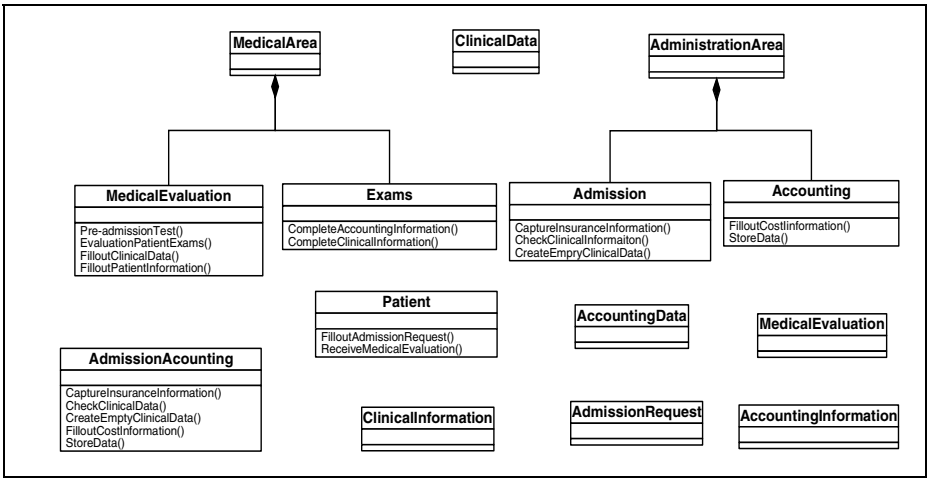

Fig. 3. Class Diagram from "Patient Admission" Business Process Specification

In Figure 4, the use case is graphically shown. This figure is enriched since, after the semi-automatic derivation, we have incorporated the use case identification, the main actor identification, the hierarchization of the areas that have subpartitions and the region denomination. In order to highlight these improvements, they have been marked with the symbol (*).

Not only analysis-level classes but also use cases complement UP at the "Requirement" and "Analysis & Design" disciplines. These artifacts form a subset of the total number of artifacts that will finally be necessary for software construction.

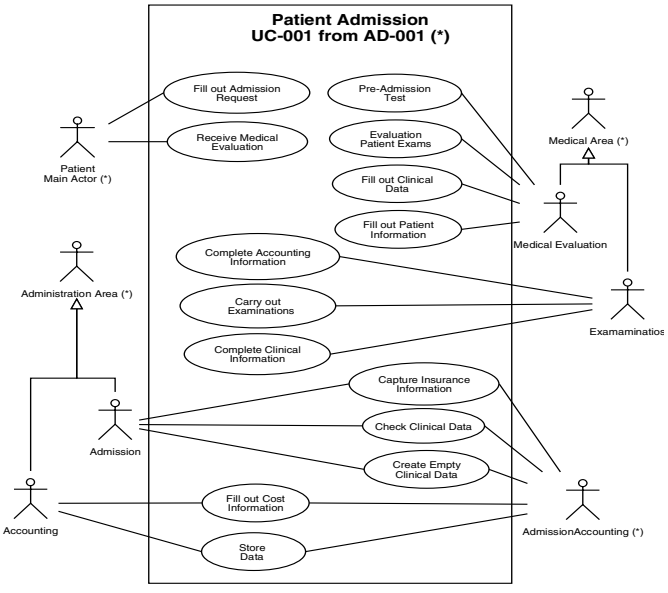

Fig. 4. Use case from "Patient Admission" Business Process Specification

5 Conclusions and Ongoing Work

The improvement presented by the UML 2.0 language and the appearance of the BPMN notation related to business process specification allow us to consider such specifications as a source of requirements to be used as an input in a software development process.

In this paper, we have presented CIM to PIM transformations in an MDA focused environment. A business process specification made by a business analyst (CIM) was used to obtain analysis-level classes and use cases (PIM). The transformation rules were specified with the use of QVT language. Both the analysis-level classes and the use cases can be used in a systematic and ordered software development process.

Ongoing work is oriented towards enriching transformations to make it possible to obtain more complete models of analysis-level classes and use cases. Our future work also has the purpose of optimizing the prototype that we have created to carry out the transformations with the aim of improving specification reuse and documentation.

Acknowledgments. This research is part of the following projects: DIMENSIONS (PBC-05-012-1), and MISTICO (PBC06-0082) both partiality supported by the FEDER and the "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha", Spain, COMPETISOFT (506PI287), granted by CYTED and ESFINGE (TIN2006-15175-C05-05/) granted by the "Dirección General de Investigación del Ministerio de Ciencia y Tecnología", Spain.

References

1. Aguilar-Savén, R.S.: Business process modelling: Review and framework. International Journal of Production Economics 90(2), 129–149 (2004)
2. Artelsmair, C., Wagner, R.: Towards a Security Engineering Process. In: The 7th World Multiconference on Systemics, Cybernetics and Informatics, vol. VI, pp. 22–27 (2003)
3. Barros, J.P., Gomes, L.: From Activity Diagrams to Class Diagrams. In: Workshop Dynamic Behaviour in UML Models: Semantic Questions In conjunction with Third International Conference on UML (2000)
4. Bézivin, J.: In Search of a Basic Principle for Model Driven Engineering. UPGRADE, European Journal for the Informatics Professional V (2), 21–24 (2004)
5. Bider, I.: Choosing Approach to Business Process Modeling - Practical Perspective, http://www.ibissoft.se/english/howto.pdf
6. BPMN: Business Process Modeling Notation (BPMN) 04.pdf (2020), http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%
7. Castela, N., Tribolet, J., Silva, A., Guerra, A.: Business Process Modeling with UML. In: 3rd. International Conference on Enterprise Information Systems, vol. 2, pp. 679–685 (2001)
8. Dijkman, R.M., Joosten, S.M.M.: An Algorithm to Derive Use Cases from Business Processes. In: 6th International Conference on Software Engineering and Applications (SEA), pp. 679–684 (2002)
9. Eriksson, H.-E., Penker, M.: Business Modeling with UML. OMG Press (2001)
10. Giaglis, G.M.: A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques. International Journal of Flexible Manufacturing Systems 13(2), 209–228 (2001)
11. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process (1999)
12. Jouault, F., Kurtev, I.: On the architectural alignment of ATL and QVT. In: ACM Symposium on Applied Computing - Model Transformation, pp. 1188–1195 (2006)
13. Kalnins, A., Barzdins, J., Celms, E.: UML Business Modeling Profile. In: Thirteenth International Conference on Information Systems Development, Advances in Theory, Practice and Education, pp. 182–194 (2004)
14. Liew, P., Kontogiannis, P., Tong, T.: A Framework for Business Model Driven Development. In: 12th International Workshop on Software Technology and Engineering Practice (STEP), pp. 47–56 (2004)
15. Lonjon, A.: Business Process Modeling and Standardization (2004), http://www.bptrends.com/
16. Object Management Group: MDA Guide Version 1.0.1, http://www.omg.org/docs/omg/03-06-01.pdf
17. OMG: Object Management Group, http://www.omg.org/
18. QVT: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. OMG Adopted Specification ptc/05-11-01 (2005)
19. Rungworawut, W., Senivongse, T.: A Guideline to Mapping Business Processes to UML Class Diagrams. WSEAS Trans. on Computers 4(11), 1526–1533 (2005)
20. Rungworawut, W., Senivongse, T.: Using Ontology Search in the Design of Class Diagram from Business Process Model. Enformatika, Transactions on Engineering, Computing and Technology 12, 165–170 (2006)
21. Štolfa, S., Vondrák, I.: A Description of Business Process Modeling as a Tool for Definition of Requirements Specification. In: Systems Integration 12th Annual International Conference, pp. 463–469 (2004)
22. White, S.A.: Process Modeling Notations and Workflow Patterns, http://www.ebpml.org/bpmn.htm