From Real Computational Independent Models to Information System Models: an MDE approach

Valeria de Castro, Juan Manuel Vara Mesa, Elisa Herrmann, Esperanza Marcos

Kybele Research Group
Rey Juan Carlos University
Tulipán S/N, 28933, Móstoles, Madrid, Spain
{valeria.decastro, juanmanuel.vara, elisa.herrmann, esperanza.marcos}@urjc.es

Abstract. Model Driven Engineering approaches deal with the provision of models, transformations between them and code generators to address software development. One of the main advantages of this approach is the definition of a conceptual structure where the models used by business managers and analysts can be traced towards more detailed models used by software developers. In this paper we focus on this kind of model transformations, that is, model transformations that allow us obtain behavioural models of the service-oriented information system (Platform Independent Models) from high level business models (Computational Independent Models). From our point of view, the main drawback regarding CIM to PIM transformations lies in the alignment between what we model at CIM and PIM levels, the business view in the former and the information system view in the later. All this given, in this work we analyze, by mean of a case study, how it is possible to integrate both views and therefore, how we can help software developers to take the most of the business knowledge represented in high level business models.

Keywords. Model Driven Engineering, Computational Independent Models, Information system Models, Model Transformations.

1 Introduction

This paper focuses on the model driven development of service-oriented information system. Service-oriented development is currently one of the main research topics in the field of software development and has brought about an evolution in Information system (IS) themselves, as well as in how they can be developed. The roadmap for research in service-oriented computing outlined by Papazoglou, Traverso, Dustdar and Leymann [roadmap] stresses the importance of defining methodologies that can facilitate the identification of meaningful services and business process specifications according to a business scenario, which are very important elements for the development of service-oriented applications [13].

According to the authors [13], one of the main challenges facing service-oriented computing is the provision of methodologies that support the specification and design of the behaviour of IS, allowing software engineers to move from the earlier stages of business analysis to the final step of implementation. However, while the design and development of simpler services is a relatively easy task, the development of complex *business services*, that is, business processes comprising several independent services, is not so simple. The main reason for this is that the transition from high-level business modelling, generally carried out by business analysts, to an executable business process which implies several software functionalities (e.g., web services, components, legacy systems, etc) is far from being a trivial issue [19]. Therefore, the problem of aligning high-level business models (corresponding to the *business view*) and information technologies (corresponding to the *information system view*) became a crucial aspect in the field of service-oriented development.

Model Driven Engineering (MDE) is an evolving and promising approach to software development [15]. MDE proposals, and more specifically its OMG specification, the MDA (Model Driven Approach) [11], constitute an important tool for the alignment of the business view and the information system view [8]. It provides with a conceptual structure that brings together the diagrams used by business managers and analysts and those used by software developers. Moreover, it is capable of organizing them in such a way that the requirements specified in one diagram can be traced (in a semi-automatic way) through other more detailed models derived from them.

In MDA, the high-level business view is represented by means of Computational Independent Models (CIM) while the information system view is represented using Platform Independent Models (PIM) and Platform Specific Models (PSM). So, attending to the alignment problem described before, in this paper we will focus on the mappings between CIM and PIM models in the framework of SOD-M [4,5], a method for the service-oriented development of IS.

To the best of our knowledge, the most of the authors working in MDE approaches do not address the already described problem of views alignment. Some of them just do not consider the CIM level of the MDA proposal, while others, the most of them, do not model the business view at CIM level. In fact, they model only the requirements of the information system, which lie into the information system view. Moreover, most of them propose using for example UML use case models [14, 10] or task diagrams [12], even when these models are not usually well known models for business experts.

In this paper, according to the MDA proposal, we claim that the CIM shouldn't be used to represent the details of the system itself, but rather those of the business domain. In that way, they serve as a bridge between the business experts and the developers of the system. The business model proposed by SOD-M, the value model [7], possesses this characteristic and opposite to others methodologies mentioned above, SOD-M incorporates a value-oriented view, which ensures that the services identified at the information system view are related to the business model of the organization using them.

In this work we analyze, by means of a case study, how we can tackle the problem of the alignment between the business view (at CIM level) and the information system view (at PIM level) following an MDE approach, taking the most of the business knowledge represented in high level business models for the information system modelling. To do so, we describe how we are actually dealing with the coding of mappings rules between both levels and we identify the main learned lessons of our work.

The rest of this paper is structured as follows. In Section 2, we briefly present the SOD-M method, which is the framework of this work. Section 3 describes the partial implementation of the case study, while section 4 presents the learned lessons obtained from it. Finally, in Section 5, we conclude by summarizing the main contributions and outlining future work.

2 Service Oriented Development-Method

SOD-M (the Service-Oriented Development Method) is a method for the service-oriented development of IS. It main features are:

– ***It defines a service-oriented approach for the development of IS*****:** it provides with guidelines for building IS based exclusively on *services* and use them as first-class objects for the whole process of the IS development. This approach facilitates the development of service-oriented applications as well as their implementation using current technologies, such as Web services [4].
– ***It is an MDA-based approach***: it proposes a set of models, extending from the CIM level, the highest level of abstraction of the MDA, to the PIM and PSM levels. Thus, by means of mapping rules between the models, SOD-M provides the benefits of the alignment of high-level business processes with the technologies currently to deploy the Service-Oriented paradigm.
– ***It defines a set of Domain Specific Languages for the service-oriented development of IS***. This set of newly DSLs includes all the modelling elements needed at the PIM and PSM levels for the modelling of IS from a service-oriented perspective at different abstraction levels. Moreover, they are based on a previously defined UML profile, thus we will be able to move from models conforming to a well-defined metamodel to extended UML models [20].

SOD-M focuses on the development of the behavioural aspect of IS and defines guidelines for building the behavioural models from high-level business modelling.

The following sub-sections present a brief description of the concepts proposed by SOD-M. Those concepts are organized according to the Business and the Information System points of view, which are integrated in order to achieve the alignment between high level business processes and information system. In this section we also present the modelling process proposed by SOD-M.

SOD-M Concepts

SOD-M proposes a new service-oriented approach for the development of IS. It considers two points of view that need to be analysed in order to develop an IS:

– ***Business View:*** it focuses on the features and the requirements of the business in which the IS will be built.

– ***Information System View:*** it focuses on the functionalities and processes that need to be implemented on the IS in order to satisfy the business requirements.

SOD-M defines a new set of concepts needed for the modelling of both views. Some of them are already known, while others are new and are proposed as part of this work. By means of a class diagram, Figure 1 shows the different concepts on which SOD-M is based, as well as their relationships.

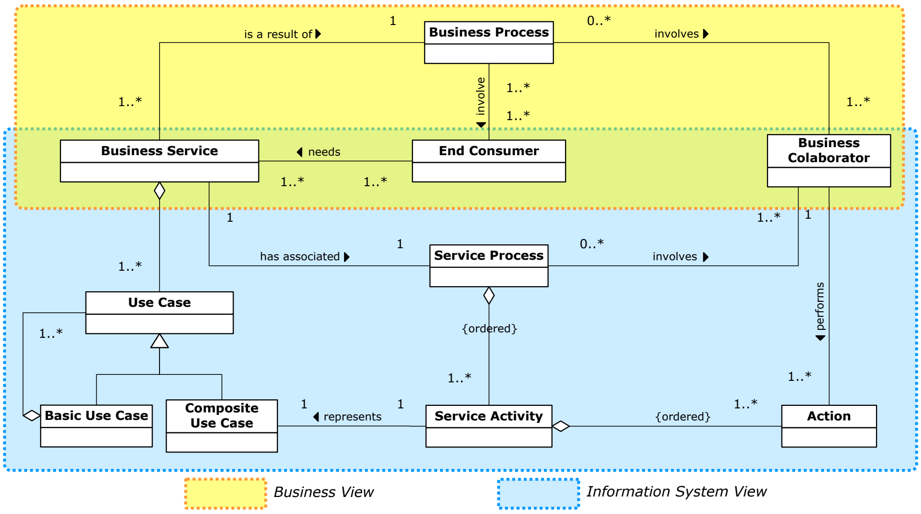

Fig. 1 SOD-M Metamodel

All of the concepts shown in Fig. 1 have a representation in some of the different models proposed by SOD-M. Firstly, those **concepts corresponding to the business view** describe the elements inherent to the business, and they will be represented on the CIM models proposed by SOD-M: the *value model* and the *business process model*. Secondly, those **concepts corresponding to the information system view** are elements used to describe the functionalities of the system, and they will be represented on the PIM and PSM models proposed by SOD-M: *use case model*, *extended use case model*, *service process model*, *service composition model*, *Web service interface model* and *Web service composition model*. Finally, those **concepts corresponding to both views** describe elements that can be analysed from both points of view, and, as we shall see later, they support the alignment between high-level business models and IS ones.

For the sake of space, we will use just the case study to describe some of these concepts. The reader is referred to [5] for more information on the rest.

The Process

The modelling process of SOD-M (see Figure 2) includes models that are in correspondence with: a) the three different abstraction levels considered by MDA

[11]: CIM, PIM and PSM, and b) the SOD-M views: the business and information system views.

As Figure 2 shows, the process begins by building the value model and the business process model, and it enables specific models for a Web service platform to be obtained as a result, thereby making it possible to make the transformation to the most typical technology related to the SOC paradigm, the Web service technology.

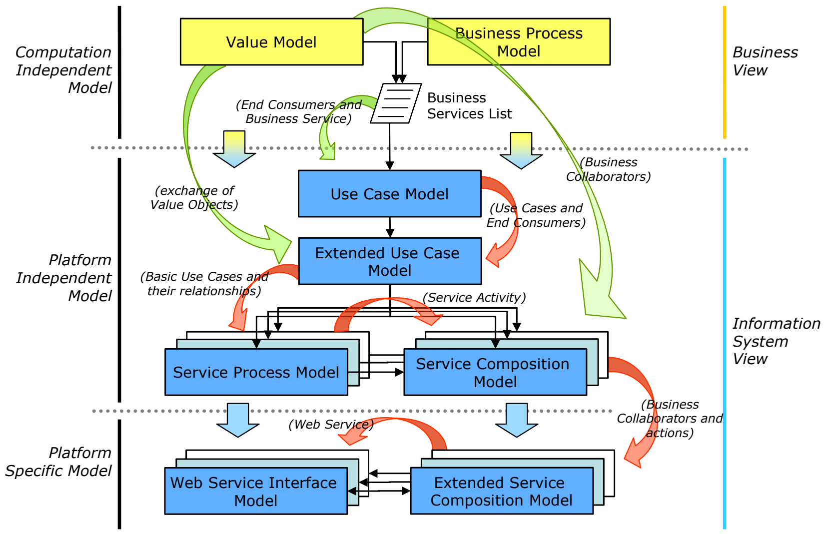

Fig. 2. SOD-M Development Process

The SOD-M process consists of several steps, each one related to the generation of a different model. From here on, we will focus on the transformation from the value model of the business view to the use case model of the information system view. We will describe how to do this transformation using the Travel Agency case study.

3 Case Study

The case study presented in this paper is a travel agency providing a set of services to its customers, these services are: accommodation, transportation and tourist packages, which include both, transportation and accommodation.

First, the customer provides a description of the required trip to the travel agency, including personal constraint and preferences. Next, the travel agency selects the Broker Agent that could serve such a trip according to the customer needs. Each Broker Agent may work with several Transportation Companies, asking them to provide with an offer for the requested service. Then, the travel agency offers the customer a set of options, according to the Customer preferences. Once the Customer selects one of the trips, he provides the credit card details to the travel agency, which processes the payment through the corresponding Financial Company. After checking that the payment is correct, the travel agency asks the corresponding Broker Agent to confirm the booking(s) and it notifies the customer.

Finally, once a month the travel agency pays each external Broker Agent for the services it has provided on the previous month.

In the next section we will describe the value model (3.1), the use case model (3.2), and the transformation between them (3.3) for this case study.

3.1 Business Modelling at CIM Level

CIM models are used to describe the environment in which the system will be used, with no direct reference to how the system will be implemented [11]. In this section we shall describe one of the CIM models proposed by SOD-M (see Figure 2): the value model.

Value Model

This model serves to depict a business case as a set of value exchanges and value activities performed by business actors. It allows us to understand the business environment in which the IS will be used and to identify some of the concepts showed in Fig. 1: the *end-consumer* of the business, the *business services* that will be offered by the system to satisfy the needs of the consumers and the *business collaborators*, those entities that collaborate in providing the business services.

The value model is obtained by applying the *e^3value* business modelling method [7], which defines a set of concepts from business administration sciences and marketing, and proposes its own specific notation for their representation in a value model. A value model shows *actors* exchanging things of economic value (*value objects*) with each other. It is also possible to represent a *market segment*, a set of actors that ascribe value to objects from an economic perspective. Furthermore, the value model shows *value activities*, activities performed by an actor who expects them to be profitable. A value activity may be, for example, a service offered to an actor.

Using the e^3value method, we have explored the business model of the Travel Agency, representing actors who wish to exchange value objects with another actor. Fig. 3 shows the value model obtained, where the travel agency is shown as an elementary actor. Customers, Financial Companies and Broker Agents are identified as market segments. The services offered by the travel agency (Accommodation, Transportation and Tourist Package) are value activities, which it expects to be profitable. The value objects are the money, the accommodation, the transportation and the tourist package, as well as the credit card verification and the fee of this service.

The value model allows the representation of *dependency paths*. Dependency paths enhance the understanding of a business idea by showing all value exchanges triggered by the occurrence of an end-consumer need. A dependency path has a direction and consists of dependency nodes. A dependency node is a start stimulus (represented by a bullet), an AND-fork or AND-join (short line), an OR-fork or OR-join (triangle), or an end node (bull's eye). A stimulus represents a trigger for the exchange of economic value objects while an end node represents a model boundary. Fig. 3 shows the dependency paths for the Travel Agency case study.

The dependency paths identified in Fig. 3 with the tag (a), (b), and (c) are initiated with the needs of the customers that are represented by start stimuli. The (a) dependency path denote a start stimulus causes by the customer when he/she requires accommodation to the travel agency in return of a payment. This payment must be checked with the financial company that gives credit card verification in return of a fee. And the travel agency gets those services from the broker agent paying an amount of money for them. Similar to (a), dependency path (c) and (b) can be described from the start stimulus to the end stimulus by means of navigating the value exchange between the actors and services.

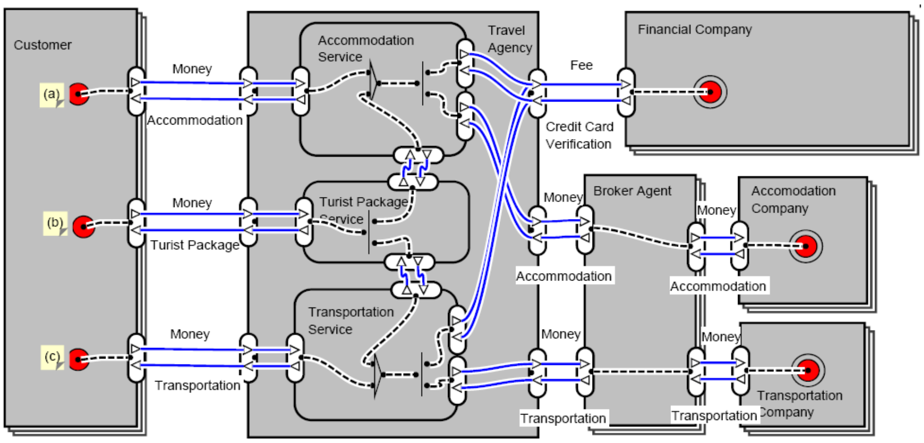

Fig.3 Value Model for the Travel Agency case study

The first thing to do when we aim to handle e^3value models in a MDE context is to define the value metamodel. To do so, we start from the e^3value ontology [7]. However, we need also to model the different value exchanges. So, we include the concepts described in the dependency path ontology in the newly metamodel.

This way, in order to automate and trace the transformation between models we define a complete value metamodel, combining both the concepts from the e^3value ontology and the dependency path technique, using the *eclipse modelling framework* (EMF) [18]. EMF is a modelling framework and code generation facility for building tools based on structured data models. Such models are specified in *ecore* format, an EMF metamodel for describing models. From an ecore (meta-)model, EMF provides runtime support for graphically editing, manipulating, reading, and serializing data based on a given (meta-)model. This way, we use EMF to build a tree-editor for models conforming to the value metamodel showed in Fig. 4 as a class diagram. We can see those classes that represent the value model in black and those that represent the dependency path in red.

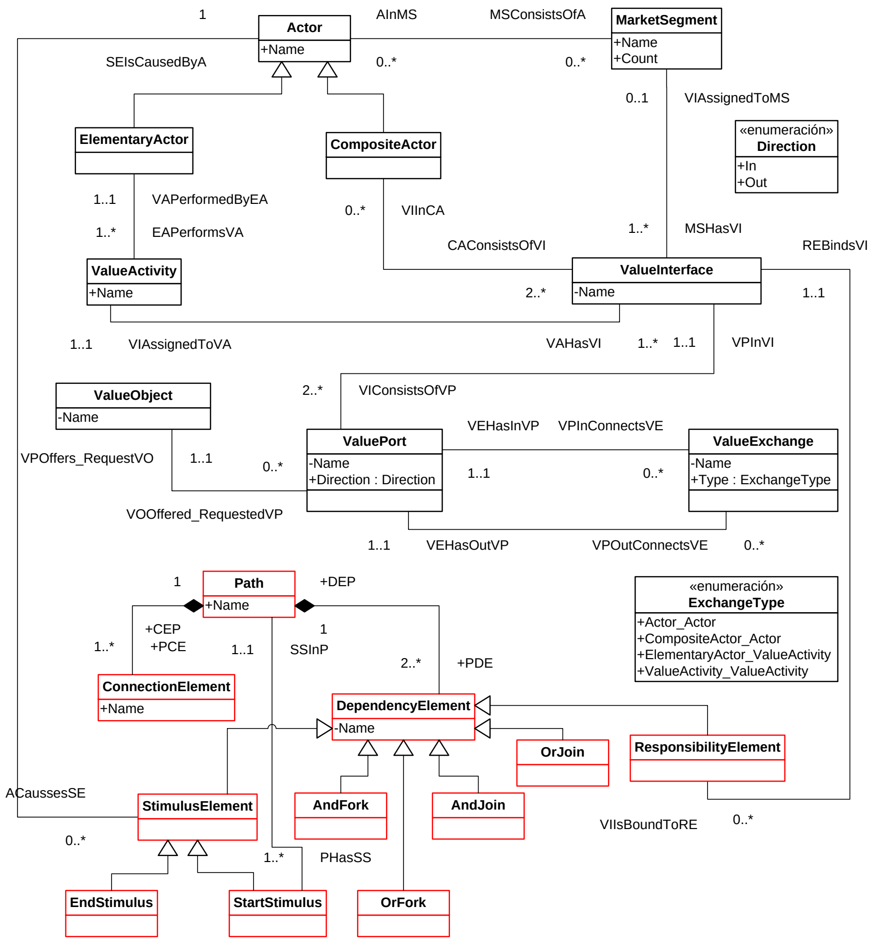

Fig.4. Value Metamodel.

Once the metamodel is specified we use EMF to generate the editor for representing models based on the e^3value methodology. Fig. 5 shows part of the value model for the case study in the generated editor. With this simple editor we are able to model different case studies, to refine those models and even more important, it is very useful when testing model transformations in order to avoid having to code XMI files by hand. However, in order to ease the task of handling value models, we are developing a graphical editor using the *Graphical Modelling Framework* (GMF) that will allow us to depict graphically value models in a UML-like way.

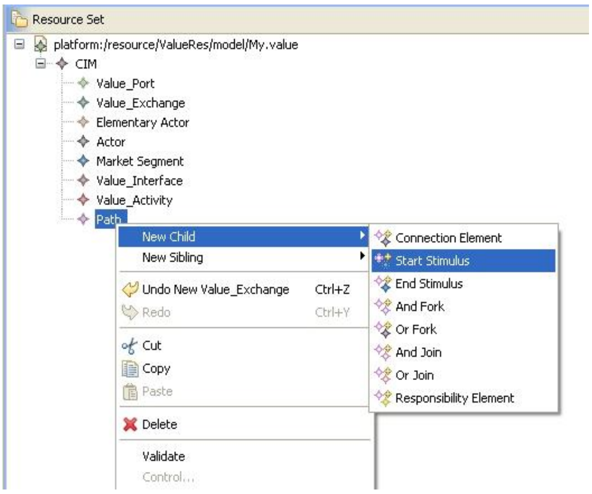

Fig.5 Excerpt of the value model for the case study.

3.2 Information System Modelling at PIM level

PIM models are used to model the functionality and structure of the information system, but without considering the technological details of the platform upon which it will be implemented [11]. SOD-M proposes different models for the behaviour modelling of the system at PIM level. Given that SOD-M defines a service-oriented approach, such models focus on the identification of the *business services* to be offered by the system, as well as the identification of the functionalities and processes needed to carry them out. Here we will describe only the *use case model* according to the topic on which this work is focused.

Use Case Model

SOD-M proposes building, as a first model of the information system, a use case model. This model only represents the *business services* to be implemented in the system, along with their relations with the *end consumers* or users of the IS. This model is represented by means of the UML use case diagram technique, but with a different building approach. In this model we identify the concepts of *end consumer*, represented as an actor; and *business service*, represented as a use case. A business service is defined in SOD-M, from an information system point of view, as that service that will be offered by the systems to satisfy a need of an end-consumer of the business.

Fig. 6 shows the use case model for the Travel Agency System (TAS), where the customer is represented as an actor who is the end consumer of the business services. The model represents as use cases the different business services that will be offered by the TAS to the customers: “obtain accommodation”, “obtain transportation” and "obtain tourist package". Such business services must be identified by taking into account the end-consumer needs identified in the value model.

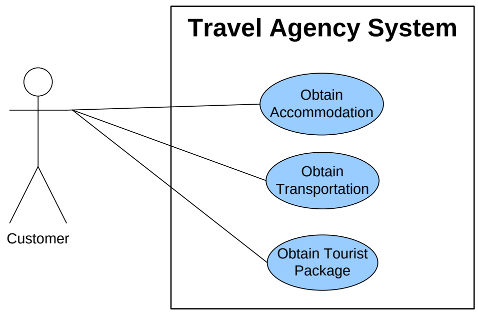

Fig.6 Use case model of TAS.

In the same way that the value metamodel was specified, we represent the use case metamodel using EMF. A simplified view of the resultant metamodel is shown in Fig. 7.

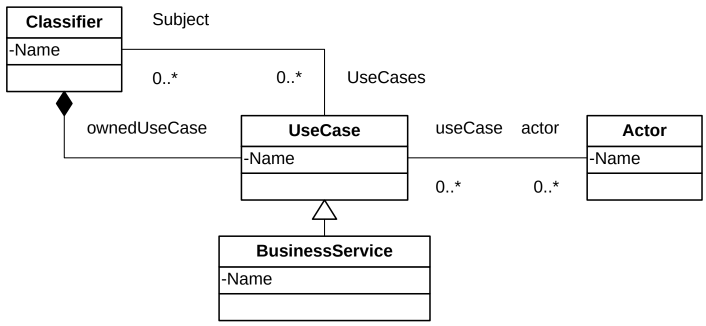

Fig. 7 Simplified Use Case Metamodel.

To handle use case models in the context of SOD-M we have already developed a graphical editor using GMF. Fig. 8 shows a screen-capture of the editor. It presents the use case model for the case study. It is obtained by applying the ATL model transformation summarized in section 3.3, taking as input the value model showed in Fig. 3.

After introducing both the value and the use case model of SOD-M, we focus on the automation of the model transformation between them.

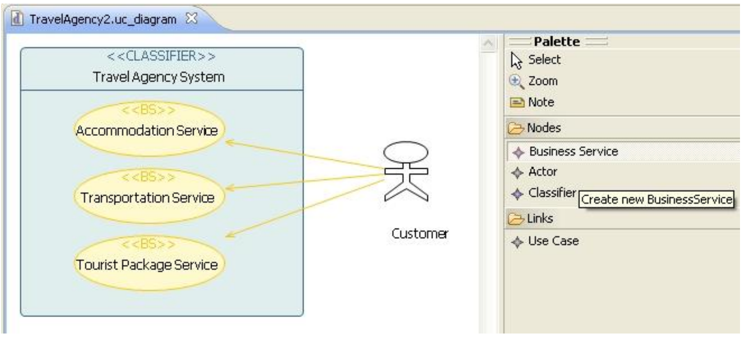

Fig. 8 Use Case Model Plug-in in order to model the case study.

3.3 Model Transformation: Value Model to Use Case Model

According to the MDA proposal, a model driven development process comprises models at different abstraction level as well as transformation between them.

Typically, MDA/MDE proposals define transformation rules between PIM and PSM models but just traceability relations between the business requirements of the CIM models and the elements of the PIM and PSM models that fulfil those requirements. This distinction comes from the nature of what each model represents: while CIM models aim to represent the business view, PIM models aim to represent those parts of the business that would be implemented in (or by) the information system.

Moving from CIM to PIM: value to use case model mapping rules

SOD-M defines guides to map requirements at the CIM level into model elements at the PIM level. These guidelines are summarized in the following table.

Table 1. Value Model to Use Case Model transformation rules.

CIM Level Value Model Elements	Mapping Rule	**PIM Level** Use Case Model Element
Actor	Identifying every actor that causes a start stimulus in the value model we obtain an end consumer of the use case model	End Consumer
Elementary Actor	Identifying on the value model the elementary actor that accomplish the value activities we obtain the classifier of the use case model	Classifier
Start Stimulus/ Dependency Path	Identifying the dependency path on the value model we obtain the business services on the use case model	Business Service
Actors/Dependency Path/Value Activity	Identifying those actors that initiate a path to certain value activity we obtain the actors that interacts with each business services	Business Service: Actor reference

Transformation Language

In order to automate the transformation we specify the transformation rules using ATL [9]. ATL (ATLAS Transformation Language) is a model transformation language and toolkit developed by the ATLAS Group (INRIA & LINA). An ATL program is basically a set of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models. ATL uses EMF to handle models: to serialize and desterilize them, to navigate and to modify them. This way, ATL works perfectly with the models defined using EMF editors.

As Fig. 9 shows, once we have identified the transformations rules to map value models into use case models, we code them using ATL. The resulting ATL program (*ValueModel2UseCaseModel*) is used to generate a use case model of the Travel Agency System from the corresponding value model. Notice that Ecore is the metamodel, not only for the source and target metamodels, but also for ATL itself.

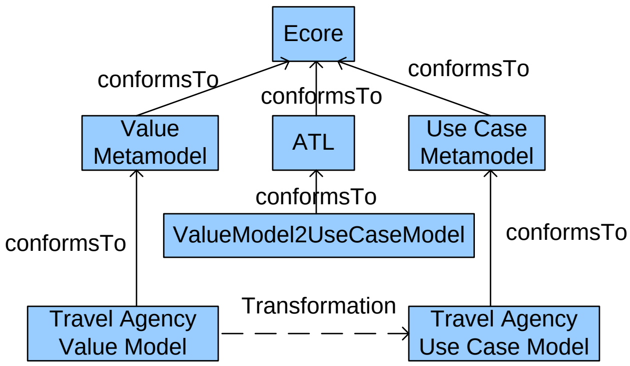

Fig.9 Case study overview

The result of the transformation is the Travel Agency use case model showed in Fig. 8. Next step would be to apply this same transformation to other case studies in order to refine them.

4 Lessons learned

As we have claimed previously in this paper, the problem of the alignment between high-level business models (corresponding to the business view) and information technologies (corresponding to the information system view) is becoming one of the crucial aspects in the area of service-oriented development.

In this paper we show how to address this problem using an MDE approach. Thus, we have defined a pair of metamodels according to the different MDA abstraction levels (business view in correspondence with the CIM level and the information system view in correspondence with the PIM level) and we have implemented the transformation between them.

After the development of this work, we can identify a set of lessons learned: some of them are related to the study of previous MDE proposals that considers CIM models, while other are related to the implementation of CIM to PIM transformations.

Regarding previous proposals we remark that the most of them don't really deal with the alignment between business and information system views. Especially since they assume that CIM models represent the requirements of the information system and not the requirements of the business in which the information system will be involved. This is a very important characteristic of the CIM models which is crucial for the development of the rest of the information system models, since they are supposed to model the details of an information system which needs to satisfy the business requirements. The CIM models included in SOD-M possess this characteristic and the main challenges for us lie in the ability to take the most of the knowledge represented in these models to (semi-)automatically generate the PIM and PSM models of the information system.

Related to the implementation of CIM to PIM model transformations we can describe the following conclusions:

a) *It is necessary to identify some kind of traceability relation* between the concepts represented at the business view, that model the business requirements and those represented at the information system view that fulfils those requirements. This traceability relation is indentified in SOD-M by means of the concepts of business services, end consumers and business collaborators (see figure 2) which can be analysed from both points of view, allowing the alignment between high-level business models and information system ones.

b) *It is necessary to establish and handle these correspondences between the elements of business and information system views, each one defined by means of a model.* These correspondences may be informal, incomplete, and preliminary. In many cases they may not be used directly to drive an executable operation. However, those correspondences can be used to drive the execution of the model transformation if we collect them using weaving models [6]. We can code a generic model transformation, which takes as input not only the source model (CIM), but also the weaving model. Then, depending on the weaving model used, we can generate different target models (PIM) for a given source model.

c) *It is possible to identify common business patterns in the business models*. Those patterns could help software engineers in the task of defining transformation rules that allow generating typical common applications, for example, common applications for travel agency management.

5 Conclusions and Future Work

The definition of methodologies that support the specification and design of the behaviour of IS, allowing software engineers to go from the earlier stages of business analysis to the final step of implementation is actually one of the main challenges to face in the context of service-oriented development. MDE approaches deal with the provision of models, transformations between them and code generators to address software development. In that way, this kind of approaches offers a conceptual structure for facilitating the alignment between high-level business models, represented at CIM level, and information system, modelled at PIM level.

In this work we have dealt with this alignment problem following an MDE approach. We have focused on the mapping between the CIM and PIM models of SOD-M, a method for the service-oriented development of IS. We have described, by means of a case study, how we are actually dealing with the implementation of the mapping rules between both abstraction levels. More specifically, we have coded the model transformation to move from a value model at CIM level to a use case model at PIM level. We have described the mapping rules needed to obtain the later from the former and we have identified the main learned lessons from this work.

At the present time we are working in the implementation of the rest of mapping rules defined in the framework of SOD-M, and more specifically those that allows us to obtain (in a way as much automatic as possible) information system details from real computational independent models. Moreover, since SOD-M follows a MDE approach for the service-oriented development of information systems, we are currently working in the code generation from SOD-M models to different Web Services platforms. All this given, we will be able to complete the alignment process between high level business models and the IT implementation.

Acknowledgements

This research is partially granted by the GOLD project financed by the Ministry of Science and Technology of Spain (Ref. TIN2005-00010) and the M-DOS (URJC-CM-2007-CET-1607) project co-financed by the University Rey Juan Carlos and the Regional Government of Madrid.

References

1. Baresi, L., Heckel, R.: Tutorial Introduction to Graph Transformation: A Software Engineering Perspective. In Corradini, A., Ehrig, H., Kreowski, H., Rozenberg, G. (eds.): Proceedings of the First international Conference on Graph Transformation. Lecture Notes in Computer Science, Vol. 2505. Springer-Verlag, (2002) 402-429.
2. Bézivin, J.: In search of a Basic Principle for Model Driven Engineering, Novatica/Upgrade Vol. 5, N° 2 (2004) 21-24.
3. Czarnecki, K., Helsen, S.: Classification of model transformation approaches. In: Bettin, J., Emde Boas, G., Agrawal, A., Willink, E., Bezivin, J. (eds): Second Workshop on Generative Techniques in the context of Model Driven Architecture (2003).
4. De Castro, V., Marcos, E., López-Sanz M.: A Model Driven Method for Service Composition Modeling: A Case Study. Int. Journal of Web Engineering and Technology. 2006 - Vol. 2, No.4 pp. 335 - 353.
5. De Castro, V. A MDA approach for the service-oriented development of WISs: From Business Model to the Web Service Composition Model. Phd Thesis. Rey Juan Carlos University, 2007. Accesible in: http://www.kybele.etsii.urjc.es/investigacion/tesis/tesis-vcastro.pdf.

6. Didonet Del Fabro, M., Bézivin, J. and Valduriez P. (2006). Weaving Models with the Eclipse AMW plugin. Eclipse Modeling Symposium, Eclipse Summit Europe, Esslingen, Germany. 2006.
7. Gordijn, J., Akkermans, J.M.: Value based requirements engineering: exploring innovative e-commerce idea. Requirements Engineering Journal Vol. 8, Nº 2 (2003) 114 -134.
8. Harmon, P.: The OMG's Model Driven Architecture and BPM. Newsletter of Business Process Trends (May 2004). Accessible in: http://www.bptrends.com/publications.cfm.
9. Jouault, F., Kurtev, I.: Transforming Models with ATL. Satellite Events at the MoDELS 2005 Conference (2006) 128-138.
10. Koch, N., Kraus, A., Cachero, C., Meliá, S., 'Integration of Business Processes in Web Application Models'. Journal of Web Engineering 3(1), pp 22-49, 2004.
11. Miller, J., Mukerji, J., 'MDA Guide Version 1.0.1' Document number omg/2003-06-01, Available at: http://www.omg.com/mda, 2003.
12. O. De Troyer, C.J. Leune, WSDM: A User Centered Design Method for Web Sites, Computer Networks and ISDN Systems 1998, 30(1-7) 85-94.
13. Papazoglou, M. Traverso, P., Dustdar, S., Leymann, F.. Service-Oriented Computing. Research Roadmap. Available at: ftp://ftp.cordis.europa.eu /pub/ist/docs/directorate_d/st-ds/services-research-roadmap_en.pdf, 2006.
14. Schmid, H.A., Rossi, G. 'Modeling and Designing Processes in E-Commerce Applications'. IEEE Internet Computing 8(1), pp. 19-27, 2004.
15. Schmidt, D.C.. Model-Driven Engineering. IEEE Computer 39 (2). February 2006.
16. Selic, B.: The pragmatics of Model-Driven development. IEEE Software Vol. 20, Nº 5 (2003) 19-25.
17. Sendall, S., Kozaczynski, W.: Model Transformation–the Heart and Soul of Model-Driven Software Development, IEEE Software archive Vol. 20, Nº 5 (2003) 42-45.
18. Steinberg, D. et al. Eclipse Modeling Framework (2nd. Ed.). Addison-Wesley Professional, 2008.
19. Verner, L.: BPM: The Promise and the Challenge. Queue of ACM Vol. 2, Nº 4 (2004) 82-91.
20. Wimmer, M, Schauerhuber, A., Strommer, M., Schwinger, M., Kappel, G. A Semi-automatic Approach for Bridging DSLs with UML, Proceedings of the 7th OOPSLA Workshop on Domain-Specific Modeling (DSM´07), Computer Science and Information System Reports, University of Jyväskylä, Finland, p. 97 – 104, 2007.