# Ontology as a Modeling Tool within Model Driven Architecture Abstraction

B. Bučko, K. Zábovská and M. Zábovský
University Science Park of the University of Žilina, Žilina, Slovakia
boris.bucko@uniza.sk, katarina.zabovska@uniza.sk, michal.zabovsky@uniza.sk

*Abstract* - **This paper is focused on automatic transformation process of top levels of Model Driven Architecture (MDA) within the information system development phase. System architects are always trying to find easier, complex and more united way of information system development. Although the Model Driven Architecture (MDA) provides a set of guidelines for the structuring of specifications it also comes with challenging tasks of transformations between the various levels of abstraction. The primary objective of this work is to design a universal automated approach within the Computer Independent Model (CIM) and Platform Independent Model (PIM) manual transformation. The manual process of the transformations within MDA could be automated using ontology model with the combination of mapping rules and Extensible Markup Process Definition Language (XPDL) and Extensible Markup Language Metadata interchange (XMI) conversion.**

***Keywords – Model driven architecture, Computer independent model, Platform independent model, Ontology, Information system development***

## I. Introduction

The process of IS development is a permanent task of software architects and developers who are trying to develop innovative solutions. On the other hand, there are the users, whose demands are challenging and lead to the exploring of new approaches of IS development. Therefore, developers are interested in creating flexible and easy to maintain information systems that can meet the requirements of users in the shortest possible time.

One of the widely used methodology among others is MDA which brings various advantages to IS development. MDA represents fundamental paradigm based on creation of models and subsequent transformations between these models. MDA specifies four levels of abstraction:

1. Computation Independent Model - CIM
2. Platform Independent Model - PIM
3. Platform Specific Model - PSM
4. Implementation Model - IM

The first three levels are graphical models, the last level is made of program code.

As was mentioned in [1] within the usage of MDA in IS development it is important to remember:

- In the first level – CIM a real (business) system of an organization is modelled and represents implementation environment of the IS that is being developed. The most common approach is to model business system with process map in some formal language. In [2] for CIM modelling there is used DFD notation, in [1] there is used Business Process Modeling Notation (BPMN). Process map is created on the basis of a consistent verbal description of the processes. PIM models are created from process map that represents CIM level model.
- According to [1] PIM level is created by two types of diagrams - use case diagram and context diagram of classes, also referred also knows as business entity diagram. Design models are created in a Unified Modelling Language (UML). A set of design patterns is a general and does not specify any particular technology. That is why it is platform independent.

According to the information above CIM level can be appropriately represented as a BPMN diagram and CIM level as a UML diagram. Additionally, the authors of [3] state that CIM level represents business viewpoint which can be easily understood even by non-IT professionals, while PIM level represents software design viewpoint.

Creating models using MDA for the development of IS in general ensures:

- systemic approach,
- complexity,
- transparency,
- methods of visual expression of IS requirements,
- methods of detecting lifecycle of the IS development.

Specific processes which should be included in the IS need a logical and efficient interconnection of data, information and knowledge which can be reached through the implementation of knowledge management. One possible approach in processing knowledge management is using of ontology.

The expression "ontology" has its origins in philosophy where it is used in the context of the question of existence and being. "What is?", "What does it mean to be?", "What is the meaning of existence?" - These questions address ontology in philosophy. It might seem that it is an abstract topic and there is no room for ontology in the IS development. Actually, that is not true. According to the definition often used by an American scientist in the field of knowledge-based systems and artificial intelligence Gruber [4] ontology is an explicit specialization of conceptualization. The term conceptualization generally expresses a system of terms and concepts which model the specific part of the world. Specification of the conceptualization must be explicitly expressed. Another definition by Gruber says that ontology is a specification represented by dictionary that is used for sharing discussions about the domain: definition of classes, relations, functions and other objects. [5]

## II. Ontology and Enterprise Architecture

The development of IS is closely related with the concept of enterprise architecture. Enterprise architecture can be characterized as a comprehensive set of key elements which create the organization. This definition, however, is considerably general. According to ISO/IEC 42010 is the definition of enterprise architecture by [6] described as follows:

Enterprise architecture is an approach, concept, medium and instruments which express the fundamental arrangement of relations between business and information system which leads to the fulfillment of the mission of the organization while respecting the surrounding environment and principles formulated in the design and development of the system.

Enterprise architecture often suffers from a lack of semantics which is reflected in issues with communication between people and between systems or even directly between people and systems. One possible solution to the problem is the enterprise architecture ontology.

Enterprise architecture ontology consists of three layers [7]:

1. Ontology of business terms.
2. Components ontology of enterprise architecture.
3. Ontology of relations between the components of the enterprise architecture.

According to authors [7], after application of this approach it is expected that humans and systems will be able to clearly and accurately understand enterprise architecture. This approach could support integration and cooperation in many organizations but also among organizations.

According to [8] ontology can be also used in the area of semantic reengineering of business processes - actually in the transformation of ontological models to non-ontological models of business processes in order to ensure interoperability between the real needs of the organization and process business models.

## III. Capturing Knowledge within MDA

Modeling CIM level within MDA is followed by transformation into PIM. The transformation process itself uncovers problematic issues relevant to information systems development which are as follows:

1. How to capture the functionality of specified information system?
2. How to transfer functional criteria directly to system design?
3. How to ensure adaptive change functionality criteria?

Existing solutions for modeling CIM level in BPMN language and subsequent transformation into the PIM level, in many cases, are not suitable for the implementation of knowledge management because these solutions cannot adequately capture the semantics of the activities performed.

BPMN modeling language in the model driven architecture is not primarily designed for modeling knowledge and semantic accuracy of created models cannot be verified. Another problem is to preserve the semantics in the transformation between levels. Knowledge modeling support in the development of knowledge IS could provide ontologies. Therefore, in further solutions we focus on using of ontologies for the development of not only knowledge-based information systems.

The CIM level in the model driven architecture expresses implementation model of IS environment. In the modern information society, regardless of the type implementation environment, CIM level reflects models of information and data, their efficient processing, switching and transforming to knowledge. All the activities of all companies and institutions are based on the collection of large amounts of information that are stored as data as well as the necessary knowledge for the operation, management and innovation. This fact thus pushes the ontology to the forefront in the field of IS development, especially knowledge-intensive IS.

According to the mentioned statements use of ontology could bring a new approach in the IS development.

## IV. Research Problem

Personalization of IS solutions can be combined with notion of enterprise architecture that will be used as a term to highlight the business connections of the organization or enterprise and developing information systems. General paradigm of complex IS development is currently the architecture of the system in accordance with ISO/IEC/ IEEE 42010: 2011 [6] and specified therein architectural framework, set up of the required views for the developing IS and the interrelationship of views. Partial views create architectures, which are generated in the relevant modeling languages.

Information systems for the promotion of knowledge management are intended to promote knowledge techniques for decision support, learning and action. The topic of knowledge-based systems is broad and contains various problems of their implementation - the overall process of development, alignment of the requirements and needs of users, applications, knowledge of different methods, integration with conventional technologies, software development tools, decision-making mechanisms, user interaction, acquisition and representation of knowledge, language and programming environment, technology implementation expertise and system architecture.

From the level perspective of model-driven architecture knowledge management as a method belongs to CIM level which in the architectural context of the development IS represents Business architecture. CIM level modeling in BPMN language and adhere to the strict rules of transformations between models CIM and PIM for modeling principles of knowledge management cannot always express the semantics of data. If we consider modeling according to the principles of knowledge management it is the modeling of information, data and knowledge. Information and knowledge belong in the architectural context to business architecture, data belong to software architecture which in the MDA represents PIM level.

If ontology can deal with relationships in CIM (information and knowledge) and the transformation of CIM-PIM (information - data) it could be more appropriate instrument for the application of the MDA principles, because it will not be necessary to create transformational relationships between CIM and PIM level.

## V. Use of Ontology as Modeling Tool within MDA

Using ontology as a modeling tool in MDA can remove the necessary transition between levels CIM and PIM using graphical modeling languages. Unification of CIM and PIM levels requires unification at the level of notation. It is therefore important in the single ontological writing (model) also select elements which are able to represent and capture relevant information from both levels.

### *A.* Modelling CIM and PIM Levels Using Ontologies

The CIM level is often expressed in terms of BPMN diagrams. In contrast, the PIM level is usually expressed in UML. Both notations (BPMN and UML) are quite extensive and comprehensive and to capture such notation would be beyond the scope of this paper. Therefore, we are focusing on several specified and frequently used elements and diagrams.

Data elements and models were extracted from the process for managing science and research at the University of Žilina. Selected BPMN elements are as follow:

- Swimming lanes,
- Pool,
- Activities,
- Process start,
- Process end,
- Decision block,
- Parallel branching,
- Timer.

Selected elements of UML diagrams use cases:

- Actors,
- Use case,
- Association.

Ontology created with the elements mentioned above should consist of the following components:

- ID element,
- Name,
- Element type,
  - Start (Start),
  - End (End),
  - Activity (Activity),

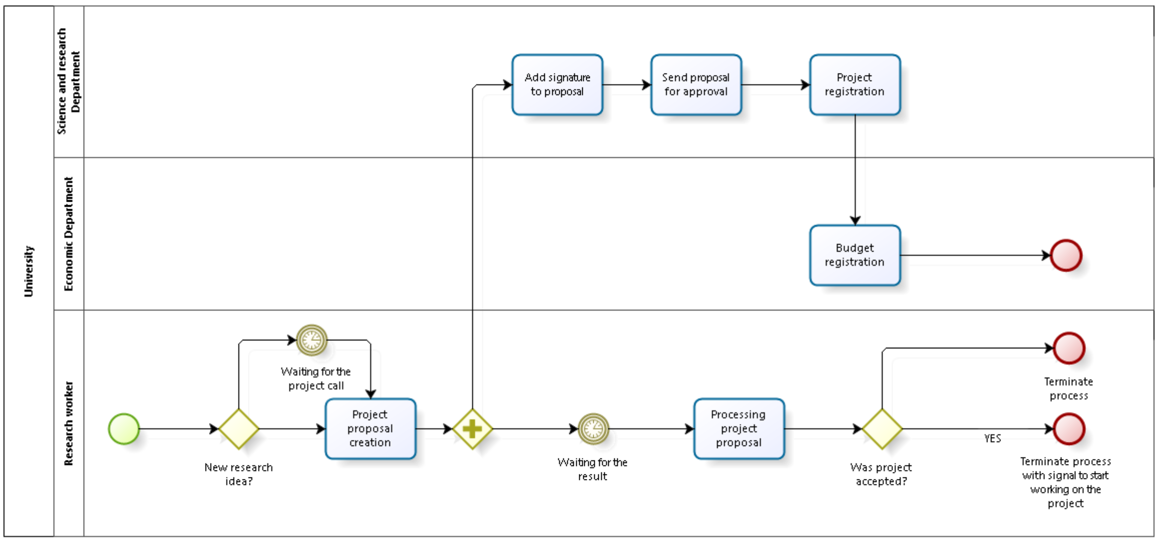


Figure 1. Selected parts from BPMN process for managing science and research at the University of Žilina – prepared within the project [9]

| | |
|---|---|
| **Element ID** | 1 |
| **Name** | - |
| **Type** | Start |
| **Source ID** | - |
| **Target ID** | 2 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 2 |
| **Name** | New research idea? |
| **Type** | Decision |
| **Source ID** | 1 |
| **Target ID** | 3, 4 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 3 |
| **Name** | Waiting for the call |
| **Type** | Timer |
| **Source ID** | 2 |
| **Target ID** | 4 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 4 |
| **Name** | Project proposal creation |
| **Type** | Activity |
| **Source ID** | 2, 3 |
| **Target ID** | 5 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 5 |
| **Name** | - |
| **Type** | Parallel |
| **Source ID** | 4 |
| **Target ID** | 6, 11 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 6 |
| **Name** | Waiting for the result |
| **Type** | Timer |
| **Source ID** | 5 |
| **Target ID** | 7 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 7 |
| **Name** | Processing project proposal |
| **Type** | Activity |
| **Source ID** | 6 |
| **Target ID** | 8 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 8 |
| **Name** | Was project accepted? |
| **Type** | decision |
| **Source ID** | 7 |
| **Target ID** | 9, 10 |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 9 |
| **Name** | Terminate process |
| **Type** | End |
| **Source ID** | 8 |
| **Target ID** | - |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 10 |
| **Name** | Signal to start working on the project |
| **Type** | End |
| **Source ID** | 8 |
| **Target ID** | - |
| **Administrator** | Research worker |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 11 |
| **Name** | Add signature to proposal |
| **Type** | Activity |
| **Source ID** | 5 |
| **Target ID** | 12 |
| **Administrator** | Science and research Dep. |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 12 |
| **Name** | Send proposal for approval |
| **Type** | Activity |
| **Source ID** | 11 |
| **Target ID** | 13 |
| **Administrator** | Science and research Dep. |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 13 |
| **Name** | Project registration |
| **Type** | decision |
| **Source ID** | 12 |
| **Target ID** | 14 |
| **Administrator** | Science and research Dep. |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 14 |
| **Name** | Budget registration |
| **Type** | Activity |
| **Source ID** | 13 |
| **Target ID** | 15 |
| **Administrator** | Economic Department |
| **Owner** | University |

| | |
|---|---|
| **Element ID** | 15 |
| **Name** | - |
| **Type** | End |
| **Source ID** | 14 |
| **Target ID** | - |
| **Administrator** | Economic Department |
| **Owner** | University |

Figure 2. Data proposal for ontological model

- Timer (Timer),
- Parallel block (Parallel)
- Decision block (Decision)

* Source ID,
* Destination ID,
* Performers,
* Owner.

Specific data are presented in the Figure 2.

## *B. Data for ontological model*

Data sample used to fulfill ontological models are created directly from the information obtained from Management of science and research at the University of Žilina. These processes have been proposed in the project Development of human resources with the support of an integrated information system for the evaluation of scientific research results [9]. Proposed data are in the Figure 2.

## *C. Ontological models*

Data presented in Figure 2 were used to create a specific ontological model written in the OWL language. Since the creation of OWL file is too large to be put clearly and shown in a whole within this paper, the sample part was selected, dealing with the element with ID 4 showed in Figure 3 below.

## *D. Mapping CIM level to ontology*

Ontological model which was briefly described in the subsection *C* can now fully cover the information that

```
<! - Http: // ... / ontologies / 2016/3 / UniverzitaOntology # 4 ->

<owl: NamedIndividual rdf: about="http://.../ontologies/2016/3/UniverzitaOntology#4">
   <rdf: type rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#Element" />
   <rdf: type>
      <owl: Restriction>
         <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasType" />
         <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#Activity" />
      </ owl: Restriction>
   </ rdf: type>
   <rdf: type>
      <owl: Restriction>
         <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasOwner" />
         <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#University" />
      </ owl: Restriction>
   </ rdf: type>
   <rdf: type>
      <owl: Class>
         <owl: intersectionOf RDF parseType="Collection">
            <owl: Restriction>
               <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasSource" />
               <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#2" />
            </ owl: Restriction>
            <owl: Restriction>
               <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasSource" />
               <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#3" />
            </ owl: Restriction>
         </ owl: intersectionOf>
      </ owl: Class>
   </ rdf: type>
   <rdf: type>
      <owl: Restriction>
         <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasTarget" />
         <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#5" />
      </ owl: Restriction>
   </ rdf: type>
   <rdf: type>
      <owl: Restriction>
         <owl: onProperty rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#hasAdministrator" />
         <owl: hasValue rdf: resource="http://.../ontologies/2016/3/UniverzitaOntology#researchWorker" />
      </ owl: Restriction>
   </ rdf: type>
   <Name>Developing the project proposal</Name>
</ owl: NamedIndividual>
```

Figure 3. Selected parts of ontological model written in OWL language

represent both CIM and PIM level of abstraction. Afterwards the mentioned ontological model can be mapped to specific CIM/PIM levels in graphical notation (BPMN or UML).

BPMN notation provides a universal graphical notation for modeling business processes. To store BPMN models we use XPDL standard (standardized by WfMC group) which allows saving of whole definition of BPMN process through specific XML files. XPDL is a generally accepted format, which is used in tools for modeling business processes.

To create a business model from ontological model we have created an application which automatically generates XPDL document describing business processes from owl file.

In order to map the ontological model to business processes with elements specified in subsection *A* – (lanes, pool, activities, process start, process end, decision block, parallel branching and the timer) it is necessary to define mapping rules and structure for XPDL file.

Mapping uses object properties defined in ontological models:

1. The swimming lane is mapped through *hasAdministrator*,
2. Pool is mapped through *hasOwner*,
3. Type mapping elements takes place through property *hasType*, the specific type may be represented by one of the following:
    a. Activity - value Activity
    b. Start - value Start
    c. End - value End
    d. Decision block - value Decision
    e. Parallel branch - value Parallel
    f. Timer - value Timer
4. Element names are mapped across the property *hasName*,
5. Mapping linking elements is solved through the features *hasSource* and *hasTarget*.

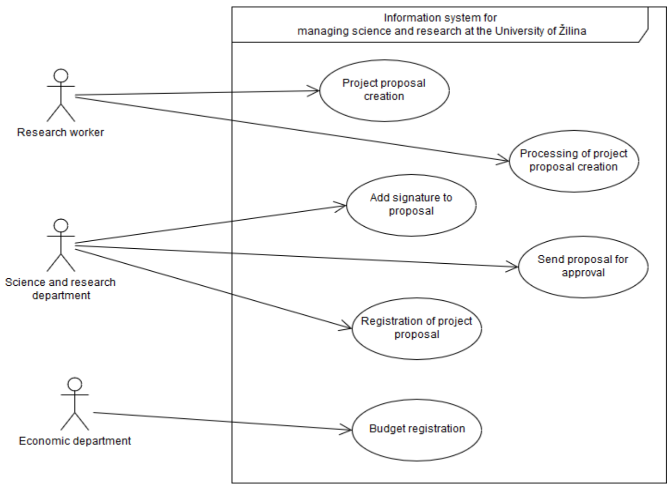


Figure 4. Use case diagram automatically generated from OWL file

XPDL structure in this case must contain the particular elements like: *Pools*, *Pool*, *Lanes*, *Lane*, *WorkflowProcesses*, *WorkflowProcess*, *Activity*, *Event*, *StartEvent*, *EndEvent*, *IntermediateIvent*, *Condition*, *Route*, *Transition*. This XPDL file can be then opened directly in various BPMN modelling tools which support XPDL file format.

### E. *Mapping PIM level to ontology*

One model that is used to represent the level of Platform Independent Model - PIM is the Use case diagram. Models expressed in UML language (including Use case diagram) are usually stored in XMI file - the generally accepted standard by the Group OMG.

In order to map the ontological model to diagram use cases with elements specified in subsection *A* - *actor*, *use case*, *association* it is necessary to define mapping rules and structure for the XMI file.

Mapping uses following object properties defined in ontological models:

1. Actors are mapped through *hasAdministrator*
2. Use cases are mapped through *hasType* - value Activity
3. Association mapping is based on the previous two points, because each activity is assigned administrator - thus it is possible to clearly determine the association.

XMI file structure can be quite complicated, involving a large number of elements and attributes. Finding an appropriate XMI structure for our purpose was carried out on the basis of experiments. After the finding the right structure we have created application that can map creation of ontological model to XMI file.

This file can be open in various modeling tools which support UML notation and allow to import the XMI format. The imported UML diagram from generated XMI file mapped from ontology is shown in Figure 4.

## VI. Conclusion

Presented solution of ontology mapping the CIM and PIM level showed how is possible to use the ontology as a modeling tool within the MDA.

Ontological model can be mapped to Computer Independent Model (CIM) and also to a Platform Independent Model (PIM). Consequently, this approach enables to represent these models in a full-fledged graphical form. In addition, the whole process of CIM-PIM modelling is much faster and there is no need for transition between levels of CIM and PIM. In order to fully eliminate the transformations, it would be necessary to extend to the mapping rules of the whole spectrum of the CIM and PIM levels in respect of any notation, restrictions, conventions, standards and file types. Nevertheless, this solution can integrate, clarify and accelerate the design, creation and maintenance of business process and use case diagrams in terms of CIM-PIM transformation and provides a platform for further development in this research area.

The advantage of the proposed solution is that any changes in the specifications of the information system proposal are made directly within ontological models and then subsequently these changes are automatically reflected through mapping into the CIM and PIM levels. Due to this fact it is not necessary to manually edit all models every time when the change within IS specification/functionality occurs.

The mentioned benefits are opening new possibilities for the development of information systems in terms of their general design (with emphasis on knowledge-based systems) and in terms of architecture and management of model transformation between CIM and PIM levels.


## ACKNOWLEDGMENT

This paper is supported by the following project: University Science Park of the University of Žilina – II. phase (ITMS: 313011D013) supported by the The Operational Programme Research and Innovation funded by the European Regional Development Fund.


## REFERENCES


[1] D. Kardoš, Tranformácia CIM do PIM v modelom riadenej architektúre (MDA), 2011.

[2] D. M. Kardoš M., "Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA),"Journal of Information and Organizational Sciences, vol. 34, no. 1, pp. 89-99, January 2010.

[3] Drozdova M., Kardos M., Kurillova Z., Bucko B. Transformation in Model Driven Architecture. In: Borzemski L., Grzech A., Świątek J., Wilimowska Z. (eds) Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part I. Advances in Intelligent Systems and Computing, vol 429. Springer, Cham, 2016.

[4] T. R. Gruber. A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2):199–220, 1993.

[5] Zabovska K., Bucko B., Zabovsky M. Using ontologies within the knowledge interpretation In: IOSR Journal of Engineering - ISSN 2278-8719. - Vol. 8, Issue 12 p. 1-3., 2018.

[6] A. Buchalcevová and L. Gála, "Architektura v podnikové informatice," Systémová Integrace ISSN 1210-9479, 15(3):7-22, 2008.

[7] Dongwoo, Jeongsoo, Sungchul and Kwangsoo, "An ontology-based Enterprise Architecture," Expert Systems with Applications, no. 37, ISSN 0957-4174, 2010.

[8] Damjanović, "Semantic reengineering of business processes," Information Systems, no. 35, 2010.

[9] University of Žilina, Development of human resources with the support of an integrated information system for the evaluation of scientific research results, ITMS Code 26110230063