Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 761–766

Procedia Computer Science

www.elsevier.com/locate/procedia

The 2nd International Workshop on the Advancements in Model Driven Engineering (AMDE)
March 23-26, 2021, Warsaw, Poland

Applying an MDA-based approach for enhancing the validation of business process models

Nemury Silega[a]* and Manuel Noguera[b]

[a]*Universidad de las Ciencias Informáticas, Carretera a San Antonio, km 2 y ½, 19370 Habana, Cuba*
[b]*Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, Pdta. Daniel Saucedo Aranda s/n, 18071 Granada, Spain*

Abstract

Business process modeling is a key activity during the development of complex and large information systems, such as enterprise management systems. These systems deal with a wide number of business processes; thus, the modeling and validation of processes becomes a challenging task. This entails dealing with issues such as the precedence between tasks and activities within a process, as well as resources, roles and enterprise assets involved. Moreover, undetected mistakes in this phase will be propagated to the system design phase and consequently will have a negative effect in the final system quality. On the other hand, the scientific literature advocates the suitability of formal models to address some issues during the process modeling. However, the adoption of formal models leads to new problems because formal languages are difficult to understand and process stakeholders usually lack of knowledge about them. In that direction, the Model-Driven Architecture (MDA) paradigm includes specifications that may alleviate some difficulties in the adoption of formal languages. Hence, in this paper we introduce an approach which combines MDA-specifications and ontologies to support process modeling. These technologies have great acceptance between both software researchers and developers. The use of ontologies permits to semantically validate the models. Furthermore, the application of MDA-guidelines could facilitate the integration of BPMN, a graphical notation for describing business process models widely accepted among business analysts, with a formal language to automate the analysis of business process models.

Keywords: MDA, ontologies and business process modeling

* Corresponding author. Tel.: +53-5322-4045
E-mail address: nsilega@gmail.com

10.1016/j.procs.2021.03.094

1. Introduction

Modern software systems are more and more large, interdisciplinary and complex, and comprise hardware, software, information, processes and facilities. In order to successfully develop these systems, the integration of holistic approaches which consider the different disciplines involved in the software development process is required. The different points of view of the involved parties (i.e., stakeholders and shareholders) also have to be considered. Enterprise resource planning (ERP) systems are an example of highly complex systems since they implement a huge number of interrelated business processes. These systems are usually interoperable, modular and settable.

Business process modeling is essential for the successful development of enterprise management systems [1]. However, it has been demonstrated that business process models usually have errors [2, 3]. These mistakes are propagated to the system design phase and consequently affect the quality of the final system. Therefore, the validation of business process models becomes a crucial activity during the life cycle of this type of systems.

There are various languages that have been widely adopted to represent business processes, such as Integrated Definition Methods (IDEF0), Unified Model Language (UML), Business Process Model and Notation (BPMN), among others. They focus on providing a graphical representation of a process that is easy to understand. However, they lack of formal semantics that allows automated model validation to be carried out. On the other hand, there exist formal languages to represent process, such as YAWL, Petri Nets and others. Some authors have also proposed the use of ontologies to represent and analyze business processes. However, the adoption of formal models leads to new problems because formal languages are difficult to understand and analysts require previous training on them. In that direction, the Model-Driven Architecture (MDA) paradigm includes specifications that may alleviate some of the difficulties in the adoption of formal languages during the process modeling.

MDA aims to tackle the complexity of software development through four main aspects: standardization, automation, raise of the abstraction level and separation of concerns [4]. MDA structures the software development process in terms of the creation and transformation of three types of models at different abstraction levels. The three types of models are: Computation Independent Models (CIMs), Platform Independent Models (PIMs) and Platform Specific Models (PSMs). As the abstraction level increase, it is improved the communication and understanding power of the model.

MDA sets three types of transformations between different levels. The first one is the CIM-to-PIM, which allows building PIMs from CIMs by adding in CIM the technical information related to system information [5]. The second one is the PIM-to-PSM, which allows adding in PIM a set of technical information related to a target platform [5]. Finally, the PSMs are transformed into executable code. Although some works deal with the CIM-to-PIM transformation [5-7], this type of transformation is not easy to achieve due to the gap between the two levels.

In this paper we introduce an approach which combines MDA specifications and ontologies to support the process modeling stage. These technologies have great acceptation in both researchers and software developers. The adoption of ontologies enables the possibilities to semantically validate the business processes. On the other hand, the application of MDA specifications could facilitate the integration of BPMN, a graphical notation widely accepted by business analysts, with a formal language to make automatic analysis. In our approach, the same information is represented with different languages for different purposes. The proposal includes a model that helps in model comprehension and a model expressed with a formal language that enables the automatic analysis. This characteristic avoids the problem declared by some authors [1] about the fact that usually the notations with good visual capabilities lack of a formal semantic, which limits the analysis of the models. On the other hand, the proposals with theoretical basis fail for the weak graphical support that hinders its comprehension by humans.

The structure of the paper is as follows. In section 2, we analyze related work. In section 3, some basic technologies of our proposal are described. In section 4, our approach is introduced. In section 5, we describe the results of a study case. Finally, section 7 presents the conclusions.

2. Related Work

Several works that remark the relevance of the business process model validation [8-11]. These works deals with different perspectives of process modeling, for example, a set of software measures [8], a guide based on patterns to manage the business process [9], rules to check the quality of the modeling [10] as well as the application of formal methods [11]. We consider the adoption of formal models as a suitable approach to validate and analyze business process models. However, since business analysts usually lack of knowledge about formal languages its adoption leads to a new challenge. The business analysts usually are experts using graphical notations to describe business processes. Hence, creating formal models from graphical models could be an alternative to reduce the complexity in the adoption of formal models. In that direction, the paradigm MDA includes specifications to generate models from other models. We found several MDA-based works [12, 13] which include models to represent and validate processes through formal models [14, 15]. Taking into account the advantages of MDA and ontologies, we have developed an MDA-approach based on ontologies to represent, validate and analyze business process models.

3. Background

3.1. Business process modeling languages

Currently, BPMN is the preferred notation for business process modelling because it defines a notation easily understandable by all stakeholders and development teams [16]. The Business Process Diagrams (BPD) are an essential part of BPMN. They diagrams include Flow Objects, Connection Objects, Swim lanes and Artifacts [17].

In spite of the utility of BPMN, it is not easy to carry out automatic validations. In order to execute automated validations it is advisable to use languages with a formal semantic. The use of ontology is a suitable option [18].

3.2. Ontology

An ontology is a formal explicit description of concepts in a domain of discourse (also called *classes*), properties about concepts describing features and attributes of the concepts (also called *slots* or *roles*), and restrictions on slots (also called *facets* or *role restrictions*) [19].

The use of ontologies could complement MDA enabling the representation of unambiguous domain vocabularies, model consistency checking and validation. Several studies argue the potential benefits of ontology to address some gaps in the process modelling [20]. OWL (Web Ontology language) [21] is one of most relevant languages for managing ontologies. OWL has significant features, such as a rich set of operators - e.g. intersection, union and negation [22]. On the other hand, with OWL it is possible to automatically check the consistency of models by means of logic reasoners.

4. An approach based on MDA and ontologies to represent and validate business processes

Our method starts with the creation of a business process model in BPMN. We chose BPMN, because it has become the standard *de facto* to represent business processes and it has a wide set of constructs to represent a BPD. However, in order to obtain a complete view of the business, it is required to represent some domain-particular concepts and restrictions, which are additional to the common elements of a process, such as activities, events and artefacts. We have named these specific concepts domain key entities (DKEs). These DKEs and restrictions help to validate the quality of the business process models automatically. For instance, according to the domain, one restriction could be the definition of some activities as mandatory for all processes.

To illustrate this we will describe a case from the enterprise management domain that we developed with the help of domain specialists. Thus, in enterprise management is it usual to deal with two concepts that we have identified as DKEs: *accounts* and *patrimonial elements*. The definition of Blanco [23] helps to understand these concepts:

"*Patrimonial elements are a set of resources, rights and payment obligations. They belong to an enterprise and represent the economic and financial instruments that allow the enterprise to reach its goals.*

Patrimonial elements are represented through an account. This instrument describes the history *of each element, it means that it is going to inform about the evolution and the value in a specific moment of each resource, rights or payment obligations because it writes with this instrument all incidences suffered in the time.*"

Therefore, *patrimonial elements* and *accounts* are DKEs in the domain of the enterprise management. Hence an enterprise management system must be able to manage them. Furthermore, we included the concept of *domain* *elements*, which will help to organize the design of an enterprise management system. These systems often include general requirements, which generate activities that are out of the business of a specific enterprise. For instance, a general requirement may be that "*all operations in an enterprise could be executed in several currencies*". This requirement demands to check the exchange rate in all operations. These requirements add restrictions to the system, are to be implemented by *domain elements*.

We have exploited the mechanism of extension that BPMN provides and which allows representing particular concepts for a specific domain in terms of BPMN. Furthermore, several tools support this functionality. We used Visual Paradigm version 13 to represent the DKEs. We have included a new attribute for the activities in order to represent the *patrimonial elements* (or *domain elements*) related to each *activity*. These DKEs will be subsequently transformed into system design entities being part of a transformation from CIM to PIM.

BPMN offers limited support to carry out automatic validation and analysis, especially in relation to DKEs. Hence, we have complemented BPMN with a formal model represented through an ontology in OWL. Since this approach is MDA-based, the ontological model is automatically created from the model in BPMN.

The classes and properties are the main components in an ontology. In order to define the classes of the ontology we have considered three main criteria: BPMN concepts, DKEs and additional concepts to represent the order of the activities in a process. According to this logic, some classes are: *Process*, *Activity*, *Event*, *Decision*, *PatrimonialElement*, etc. To represent the order of the activities [24] we included the classes *Step*, *StartStep*, *EndStep*, *IntermediateStep*, *FlowElements*. The ontology also includes properties which link the different concepts.

Table 1 shows the rules to translate the each construct in BPMN into an ontological entity. The rules are described in natural language, but we have specified them by using the Atlas Transformation Language (ATL) too.

Table 1. Transformation rules

No	BPM concept	Concept in the ontology
1	Activity	Instances of the classes *Activity* and *Step* and a Statement to relate the *Step* and the *Activity*
2	Type	Instance of the class *Type*
3	Patrimonial Element	Instance of the class *PatrimonialElement*
4	Domain Element	Instance of the class *DomainElement*
5	Event	Instance of the class *Event*
6	Gateway	Instance of the class *Gateway*

We have used the ATL plugin for Eclipse in order to specify the transformation rules. Fig. 1 a) shows the structure of the ATL project; each file is explained below:

- **Metamodels**:
  - OWL.ecore: OWL Metamodel in Ecore EMF XMI 2.0 format.
  - XML.ecore: XML metamodel in EMF XMI 2.0 format.
- **Transformations**:
  - BPM2O_BPM.atl: It generates a process model represented in OWL from a process model represented in BPMN. Fig. 1 b) shows an excerpt of rule 1 of Table 1. In this transformation, three elements are created in the ontology from each task in the BPMN model. From line 630 to line 643 an instance of the class *Activity* is created for each task in the BPM. From line 645 to line 656 an instance of class *Step* is created. Finally, a statement is created in order to associate the *Activity* and the *Step* by means of the object property *Executed_To*.
  - OWL2XML.atl: It translates an OWL model into an XML model with OWL/XML syntax elements. This is an extra transformation in order to represent the ontologies in a format that could be interpreted by OWL editors, i.e. Protégé. We reused this transformation from the web page of ATL

- **Models**: This folder has two subfolders (**In** and **Out**). In the folder **In** the input models are located while in the folder **Out** the generated models are located.

We developed an ant script (*buld.xml* file) in order to execute the transformation chain. Therefore, the transformations can be executed automatically in one-step.

Fig. 1. a) Structure of the ATL project and b) Excerpt of the rule with ATL language Task2Activity

5. Case Study

In order to illustrate the application of our approach, we used the *Settle advanced payments* process. The flow starts when a finance officer checks the file of a provider who makes a payment; he/she collates the invoice with the anticipated payment and considers the data of the purchase. Then, the officer updates the provider´s file and sends the operation for accountability. Fig. 2 a) shows the model.

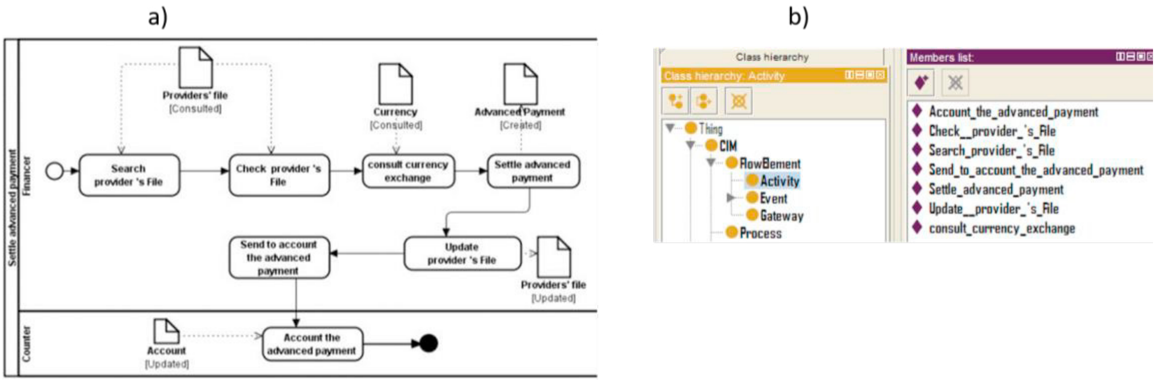

Fig. 2. a) Diagram of the process *Settle advanced payment* and b) Generated Activities in the ontology.

In addition to the elements which are graphically showed in Fig. 2 a), we included the information related to the DKEs in the models. For example, the Patrimonial elements considered in this process are *Account* and *Advanced payment*, while *Currency* is a Domain element. Therefore, we specified in the model that the activity *Account the advanced payment* is related to the Patrimonial element *Account*; the activity *Settle advanced payment* is related to the Patrimonial element Advanced payment and the activity *Consult currency exchange* is related to *Currency*.

After executing the transformation BPM2O_BPM.atl with the process model introduced above as input, we obtained the ontological model describing the process *Settle advanced payments*. Figure 2 b) shows generated instances of the class *Activity*. Likewise, the remainder elements of the business process model are added to the ontology. Then, it is possible to validate the process model with a reasoner and check the quality of the models. We added mechanisms to detect errors related to DKEs in the ontology.

6. Conclusions

Business process modeling can take advantage of ontology-driven engineering methodologies and tool support in order to ensure models consistency and detection of errors in early stages of software construction. We have presented an approach based on ontologies and illustrated its application to the domain of enterprise management. The method makes use of ATL for translating models in BPMN into a formal model represented in ontologies. This approach helps bridge the gap between graphical notations and formal models to represent and validate business processes. Despite the fact that analysts prefer graphical notations to describe business processes, formal models are better for automatic analysis. Hence, this approach takes advantages of the two types of models. Furthermore, the direct transformation ensures the consistency between both models and reduces the efforts of the analysts.

Acknowledgements

This research has been partially funded by the Dpto. de Lenguajes y Sistemas Informáticos from the University of Granada. Authors also acknowledge input from COST Action CA19134 - Distributed Knowledge Graphs.

References

[1] J. Barjis, "The importance of business process modeling in software systems design," *Science of Computer Programming,* vol. 71, pp. 73-87, 2008.
[2] J. Mendling, "Empirical Studies in Process Model Verification," in *Transactions on Petri Nets and Other Models of Concurrency II*. vol. 5460, K. Jensen and W. M. P. van der Aalst, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 208-224.
[3] I. Moreno, M. Snoeck, H. A. Reijers, and A. Rodríguez, "A systematic literature review of studies on business process modeling quality," *Information and Software Technology,* 2014.
[4] B. Selic, "Manifestaciones sobre MDA," *Novática,* pp. 13-16, 2008.
[5] Y. Rhazali, Y. Hadi, I. Chana, M. Lahmer, and A. Rhattoy, "A model transformation in model driven architecture from business model to web model," *IAENG International Journal of Computer Science,* vol. 45, pp. 104-117, 2018.
[6] Y. Rhazali, Y. Hadi, and A. Mouloudi, "Model Transformation with ATL into MDA from CIM to PIM Structured through MVC," *Procedia Computer Science,* vol. 83, pp. 1096-1101, 2016/01/01/ 2016.
[7] N. Silega, M. Noguera, and D. Macias, "Ontology-based Transformation from CIM to PIM," *IEEE Latin America Transactions,* vol. 14, pp. 4156-4165, 2016.
[8] E. Rolán, F. Ruiz, F. A. G. Rubio, and M. Piattini, "Aplicación de métricas software en la evaluación de modelos de procesos de negocio," *Revista Sociedad Chilena de Ciencia de la Computación,* vol. 6, 2005.
[9] P. Bonillo, "Metodología para la gerencia de los procesos del negocio sustenda en el uso de patrones," *Journal of Information Systems and Technology Management,* vol. 3, pp. 143-162, 2006.
[10] S. Díaz-Montenegro Quesnel, "Metodología de definición de procesos," Univ. Politécnica de Madrid, Facultad de Informática, 2009.
[11] J. A. Fisteus, "Definición de un modelo para la verificación formal de procesos de negocio," Phd Thesis, Univ. Carlos III de Madrid, 2005.
[12] M. Melouk et al., "An Approach for Transforming CIM to PIM up To PSM in MDA" *Procedia Computer Science,* vol. 170, pp. 869-874, 2020.
[13] N. Kharmoum, S. Ziti, Y. Rhazali, and O. Fouzia, "A Method of Model Transformation in MDA Approach from E3value Model to BPMN2 Diagrams in CIM Level," *IAENG International Journal of Computer Science,* vol. 46, 2019.
[14] Z. Li, X. Zhou, and Z. Ye, "A Formalization Model Transformation Approach on Workflow Automatic Execution from CIM Level to PIM Level," *International Journal of Software Engineering and Knowledge Engineering,* vol. 29, pp. 1179-1217, 2019.
[15] N. Laaz, K. Wakil, Z. Gotti, S. Gotti, and S. Mbarki, "Integrating Domain Ontologies in an MDA-Based Development Process of e-Health Management Systems at the CIM Level," in *Intl. Conf. on Advanced Intelligent Systems for Sustainable Development*, 2019, pp. 213-223.
[16] OMG, "Business Process Model and Notation (BPMN). Versión 1.2", OMG2009.
[17] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard" *Computer Standards & Interfaces,* vol. 34, pp. 124-134, 2012.
[18] W3C, Jeff Z. Pan et al., "*Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineerin"g*. Available: http://www.w3.org/2001/sw/BestPractices/SE/ODA/.
[19] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology" Stanford Medical Informatics, Stanford SMI-2001-0880, 2001.
[20] J. B. Gassen et al., "An experiment on an Ontology-Based Approach for Process Modeling," *Information and Software Technology,* 2016.
[21] J. Xing and T. Ah-Hwee, "CRCTOL: A semantic-based domain ontology learning system," *Journal of the American Society for Information Science & Technology,* vol. 61, pp. 150-168, 2010.
[22] M. Horridge, "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.2" The University Of Manchester, Manchester 2009.
[23] R. E. R. Blanco. (2008). *Contabilidad y fiscalidad*. Available: www.eumed.net/libros/2008b/396/.
[24] M. Noguera et al., "Ontology-driven analysis of UML-based collaborative processes using OWL-DL and CPN," *Science of Computer Programming,* vol. 75, pp. 726-760, 2010.