HAL
open science

Translation of heterogenous requirements engineering meta-models

Imad Eddine Saidi, Mahmoud El Hamlaoui

To cite this version:

Imad Eddine Saidi, Mahmoud El Hamlaoui. Translation of heterogenous requirements engineering meta-models. 6th International Workshop on Requirements Engineering (IWoRE 2013), Oct 2013, Constantine, Algeria. pp. 1-5. hal-01188596v2

HAL Id: hal-01188596

https://hal.science/hal-01188596v2

Submitted on 11 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open Archive TOULOUSE Archive Ouverte (OATAO)
OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : http://oatao.univ-toulouse.fr/
Eprints ID : 12960

To cite this version : Saidi, Imad Eddine and El Hamlaoui, Mahmoud *Translation of heterogenous requirements engineering meta-models.* (2013) In: 6th International Workshop on Requirements Engineering (IWoRE 2013), 19 October 2013 - 20 October 2013 (Constantine, Algeria). (Unpublished)

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Translation of Heterogenous Requirements Engineering Meta-Models

Study Case : Automated Teller Machine

SAIDI Imad Eddine
University of Constantine 2, LIRE Laboratory,
SIBC Team, Constantine, Algeria
imad-eddine.saidi@irit.fr

EL HAMLAOUI Mahmoud
University of Med V Souissi ENSIAS, SIME Laboratory,
IMS Team, Rabat, Morocco
mahmoud.hamlaoui@um5s.net.ma

***Abstract*— The globalization and the rapid development of information and communication technologies encourage organizations to work together. In software development, many works have emerged to support this cooperation using different tools and methodologies. Most of them focus on the design-stage concerns. However, very little works have dealt with cooperation during the early stage of software projects, namely Requirements Engineering (RE), despite the importance of this stage for the failure or the success of software projects. There exist different kinds of approaches to support the RE process in different contexts, based on models such as goal, viewpoint and scenario oriented. Each of these models relies on concepts which differ from one model to another. We propose a Unified Requirements Engineering Meta-model (UREM) which allows cooperation in the requirements engineering process between heterogeneous RE models. In this paper, we explore UREM as a pivot meta-model which performs translation between different RE models in order to ensure interoperability between heterogeneous RE models.**

Keywords-component; Requirements Engineering; Pivot Model; styling; insert (key words)

I. INTRODUCTION

In Requirements engineering, companies have different cultures and use different kinds of tools and approaches to describe and manage upstream phases of software projects such as goal-oriented, viewpoint-oriented and scenario-oriented approaches. The globalization and the rapid development of Information Technologies sometimes require companies to work together in various fields including RE in order to achieve common objectives as quickly as possible. Another thing, these companies are not ready to agree on a unique RE approach to cooperate because of the time and the cost that result from the migration. The aim of the RE meta-model (UREM) proposed as an intermediary of communication between different types of meta-models of RE approaches in order to allow cooperation between these approaches.

Bendjenna (Bendjenna and al, 2010) has proposed an integrated approach MAMIE which combines different kinds of concepts: goal, scenario and viewpoint in order to allow cooperation between companies. In i* approach, there exists different variations for particulars usages. Carlos (Carlos and al, 2011) has defined super meta-model hosting identified variations of i* and implementing a translation algorithm between these different variations oriented to semantic preservation. Our work intends to be a combination between the two works. We propose an abstract meta-model which allows cooperation and translation of information between different kinds of RE approaches.

This paper is organized in six sections. In section two we present an overview of the idea behind UREM. In section three, we present our three-steps unification process. In section four, we draw UREM. In section five, we discuss the performance of the implementation of the unification process. Finally, we conclude and draw perspectives of this work.

II. INTEROPERABILITY OF HETEROGENEOUS MODELS

Conceptually, the interoperability of heterogeneous models can be performed either directly without intermediate transformation or after a transformation tin order to express models within the same "pivot" language.

A pivot model is a model used as an intermediate representation to align the input models to the same formalism.

The concept of a pivot model has been introduced in several research areas related to the model-driven engineering especially in taking into account the interoperability.

Commonly, the term “pivot” means the point of rotation in a lever system. It is also the term used to describe an interpreter who translates a low level language “Maltese” (national language of Malta) to a language (e.g : English). The translated text is then used as a source of translation for other languages (Beleg and al., 2009).

One of the first uses of the term “pivot” in Computer Science referred to the quicksort algorithm numbers (quicksort). The algorithm consists in choosing a number (called pivot) from a list of disordered numbers and switch all the elements, so that all those who have a lower value to the pivot are placed to the left and all those who a higher value on his right.

Milanovic and al. introduced in (Milanovic and al., 2009): R2ML (Rewerse Rule Markup Language), a pivot metamodel for bidirectional alignment taking into account in one hand the ontologies that are the backbone of the semantic web and in the other hand MDA concepts.

In the field of ontologies, central area of the Semantic Web, the models are described by OWL (Ontology Web Language) and SWRL language (Semantic Web Rule Language) for expressing validation rules for the semantic web. Whereas in the field MDA, models are described in UML with OCL as constraints expression language.

Thus models expressed in UML / OCL can be exploited in the field of semantic web by translating them into OWL / SWRL models and vice versa through neutral model: R2ML.

Similarly Sun and al. in (Yu Sun and al., 2009) have defined a pivot model. Many tools according to them are developed to automatically detect redundant codes in a program and represent them into appropriate statistics. The problem is that each of these tools has a different representation of the obtained result which gives the integrator a hard task to know each of these representations in order to act on the portion of the appropriate program.

The idea presented is to provide a common graphical representation in SVG. This is achieved by defining a meta-model pivot GCC (Generic Code Clone) that contains common concepts and characteristics of redundant code blocks detection tools. This meta-model will serves as (intermediate) between redundant code detection tools and SVG (Scalable Vector Graphics) model.

The use of pivot as an intermediate model makes it easy to centralize and optimize the data format in order to represent the input models in the same formalism. The interoperability process is thus simplified and can continue with less complexity.

Figure below, illustrates the role of the pivot to perform translation between heterogeneous models.

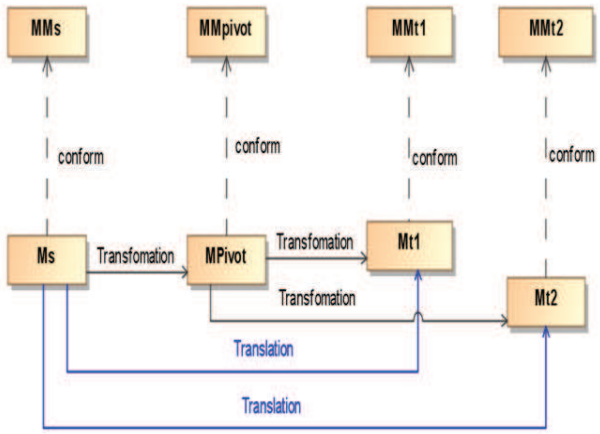

Figure 1. Translation between modes using a pivot model

Ms, Mt1 and mt2 are different models that are conform to the following metamodels MMs, MMt1 and MMt2.

In our case we need to translate Ms into Mt1 and Mt2, this translation cannot be done directly because the source models are heterogeneous. To solve this issue, we define a translation from the source model to a predefined pivot model (Mpivot). Once the model is available we could target the needed model through another transformation. Those transformations are easy to establish due to the fact that the input models are known.

In the next section we explore our Pivot model (UREM) which is proposed to perform translation between different RE models.

III. Unified Requirements Engineering Meta-Model

UREM is an intermediary of communication and information translation between different types of RE models.

RE models are instances of different types of RE Meta-Models where each Meta-Model is composed of a set of concepts.

The idea behind the pivot UREM is to create a new Meta-Model which is composed of a set of classes where each class is an abstraction of a set of concepts (similar concepts) that exist in different RE Meta-Models.

To find abstractions between RE concepts, we have adopted a rigorous process that is concerned with the meaning of concepts (Semantic Process).

Our process is based on WordNet (George, 1995) to find semantic relationships and similarities between words which represent RE concepts (words are the only thing that we get to apprehend RE concepts).

Our aim is to perform cooperation between different types of approaches. In the unification process, we have chosen one approach from each type of RE approaches in order to achieve our goal, regardless of the RE approach chosen, our unification process is applicable to various other approaches. In this paper, we deal with approaches that are widely used: i* (Castro and al, 2011) as goal oriented approach, CREWS (Sutcliffe and al, 1998) as scenario oriented approach and PREview (Sommerville and al, 1997) as viewpoint oriented approach.

The resulted Meta-Model UREM is shown in figure one.

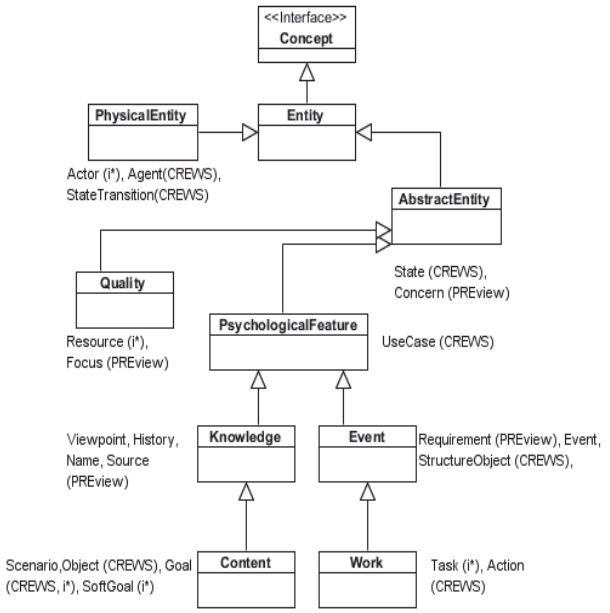

Figure 2. Unified Requirements Engineering Meta-Model

IV. DEDUCTION OF TRANSLATION RULES FROM UREM

In this section, we deduct translation rules from UREM illustrated in figure one. We observe in figure one a list of concepts of different RE Meta-Models near each class of UREM. So, each class covers a set of concepts and plays the role of a pivot between these concepts. Proceeding from UREM, we are looking to find for each source concept c_S of model M_S, a target concept or a set of target concepts c_T of model M_T. We perform two-way translation between two given models M_1 and M_2, first from M_1 to M_2 then we translate each not-translated concept of M_2 to a target concept of M_2. Two-way translation allows us to ensure that we have applied translation on all concepts of all different RE models.

We conclude two types of translation between concepts: Direct Translation and Inheritance Translation. The following sub-sections describe each type of translation.

A. Direct Translation

This type of translation is used if the source concept c_S of a model M_1 and the target concept c_2 of a model M_2 share the same abstraction c_G in UREM. To perform translation from the source concept to the target concept, we check the abstraction of the source concept then create the target concept by implementing the abstraction c_G in two steps. First, copy shared attributes to the target concept then translate the rest of attributes one by one.

For example, the concept *Resource* in an instance of i* meta-model share the same abstraction *Quality* in UREM with the concept *Concern* of a PREview model then *Resource* concept must be translated to a *Concern* in PREview model and vice versa.

B. Inheritance Translatioin

This type of translation is used if a source concept c_S of a model M_S can't be translated to any target concept c_T of model M_T using Direct Translation. In this type of translation, we check classes that are linked to the abstraction c_G of the source concept in order to find the abstraction of a target concept c_2. Since we care more about details of concepts, we check first child classes of c_G. If no abstraction of a target concept is founded, we check parent classes (abstractions of c_G). If no abstraction of a target concept is founded then the source concept c_S can't be translated to a target concept c_G.

For example, to perform translation from a *Requirement* of a PREview model to other concept in i* model. We observe that Requirement doesn't share an abstraction which is *Event* with any concepts of i*. Thus, we look up child classes of *Event* class level by level. We find that the child class *Work* of *Event* covers the concept *Task* in i* then, when we perform translation from a PREview model to an i* model, the concept *Requirement* of PREview must be translated to the concept *Task* of i* and vice versa.

The activity diagram which describes the overall process of finding translation between concepts is shown in figure

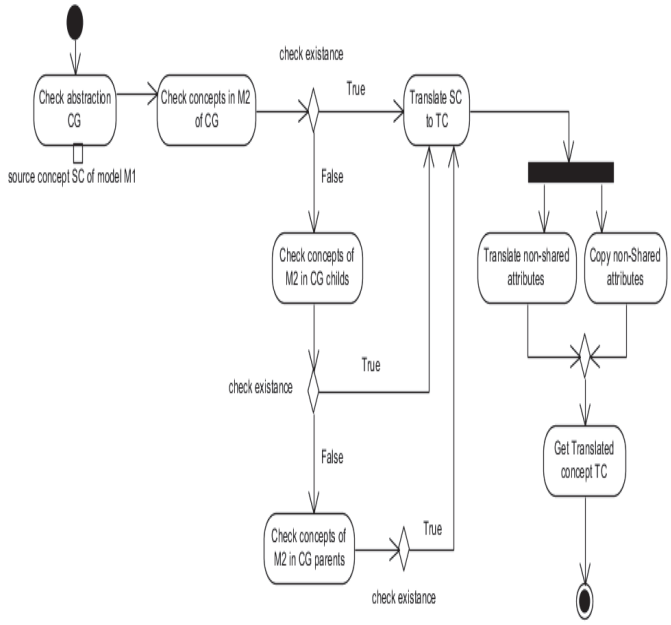

Figure 3. Activity diagram of translation between requirements engineering concepts

V. CONCLUSION

This paper has presented a unified requirements engineering meta-model that is resulted from a semantic unification process of different requirements engineering meta-models. The unification process is based on finding semantic similarities between different concepts that already exist in different types of requirements engineering meta-models. The aim of the unified requirements engineering meta-model UREM as mentioned in section two and three is to perform

translation between different types of requirements engineering models in order to allow cooperation between companies that have different cultures and use different kinds of Requirements Engineering approaches. The translation rules are deducted in section four directly from the unified Meta-Mode UREM. One of the gaps of these translation rules is the lack of concrete semantic translation at attributes level of concepts. We seek to fix this issue of semantic translation between concepts in a future work. We seek also to demonstrate other features of UREM such as evolution and composition. Evolution is how UREM is easy to update and maintain. Composition is how to compose a full requirements specification document of a project from different pieces of requirements specifications arisen from different models. Afterward, we are looking to implement a visualization tool in order to present and illustrate the translation operation between models as graphs.

Bendjenna, H., Zarour, N.E., Charrel, P.J., 2010. Eliciting Requirements for an inter-company cooperative information System. *Journal of Systems and Information Technology (JSIT)*,

Carlos, C., Xavier, F., 2011. A Metamodelling Approach for i* Model Translations. *23rd International Conference, CAiSE 2011*.

G. Beleg. Design and prototypical implementation of a pivot model as exchange format for models and metamodels in a qvt/ocl development environment. Ph.thesis, Université de technology Dresden, 2007.

Kazmierczak, E., 2003. Requirements Engineering 433-641E, Course-work Masters The University of Melbourne First Semester.

M. Milanovic, D. Gaševic, A. Giurca, G. Wagner, and V. Devedžic. On interchanging between owl/swrl and uml/ocl. In Proceedings of 6th Workshop on OCL for (Meta-) Models in Multiple Application Domains (OCLApps) at the 9th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS), Genoa, Italy, pages 81–95, 2006.

Yu Sun, Zekai Demirezen, Frédéric Jouault, Robert Tairas, and Jeff Gray. Software language engineering. chapter A Model Engineering Approach to Tool Interoperability, pages 178–187. Springer, 2009.