

# UMLtoSBVR: An SBVR-based tool to validate UML conceptual schemas

Raquel Pau <sup>1</sup>

Jordi Cabot <sup>2</sup>

Ruth Raventós <sup>1</sup>

**Abstract:** The UMLtoSBVR tool facilitates the interaction between designers and business people in order to refine and validate the information modeled in a UML conceptual schema (CS) to make sure that the CS is correct before starting the implementation phase. To this end, the tool is able to paraphrase the CS, i.e. to describe the CS elements using natural language expressions that can be understood by the business people lacking of the technical background to directly understand the UML notation. As an intermediate step, the tool transforms the UML CS into a SBVR (Semantics of Business Vocabulary and Business Rules) specification. The SBVR standard facilitates the expression of the CS in different natural language styles (as Structured English or Rule Speak).

---

<sup>1</sup> Universitat Politècnica de Catalunya (UPC) [raquelpau@gmail.com](mailto:raquelpau@gmail.com), [ruth.raventos@upc.edu](mailto:ruth.raventos@upc.edu)

<sup>2</sup> Estudis d'Informàtica, Multimedia i Telecomunicació, Universitat Oberta de Catalunya  
[jcabot@uoc.edu](mailto:jcabot@uoc.edu)

## 1 Introduction

The specification of an information system (IS) must include a formal representation of the knowledge of the domain required by the IS to perform its functions. In conceptual modeling, this representation is known as the conceptual schema (CS) of the IS. A CS is the primary artifact of the development process and it must include all relevant dynamic and static aspects of the domain, including all business rules. CSs are described in a particular conceptual modeling language. Nowadays, the Unified Modeling Language (UML) is the most widely used modeling language for this purpose.

Before proceeding with the implementation of the IS, designers must establish the correctness of the CS they have specified. This is even more important in a model-driven development context, where the quality of such CS has a direct impact on the quality of the final system implementation (that is semi-automatically generated from it). This implies that the CS should be verified (to be sure designers “defined the CS right”) and validated (to be sure designers “defined the right CS”). Verification of UML schemas has been largely addressed but, unfortunately, validation can neither be achieved by purely automatic methods nor by designers themselves. In fact, validation requires that designers and business people reach an agreement on what the system is to perform. However, it is unlikely that stakeholders are able to directly understand a CS represented in UML.

Our UMLtoSBVR [2] tool implements one of the possible solutions to this problem, suggested by *Sommerville* and *Sawyer* [1] and consisting in paraphrasing the CS (i.e. describing its elements by means of natural language expressions) so that stakeholders can understand and validate the knowledge expressed in the CS before the implementation phase of the IS starts. Given an UML CS as input, the tool generates a (human-readable) textual description of the CS as an output.

Our tool relies on the recent *Semantics of Business Vocabulary and Business Rules* (SBVR) specification [3] as an intermediate representation. SBVR is targeted to capture business concepts and business rules in a language close enough to ordinary language to permit business experts to read them, and at the same time formal enough to be suitable for communication among organizations or software tools. Thus, SBVR is specially suited for acting as an intermediate representation between the stakeholders and the designers.

Therefore, and following the method presented in [4], the tool is implemented as two model transformations, a first model-to-model transformation that generates a SBVR representation from the input UML CS and a second model-to-text transformation that generates a textual (natural language) description of that SBVR representation. Having two separate transformations permits to reuse the first one in other application scenarios and easily adapt the second one to the specific language/style desired by the business people without having to change the first part of the transformation chain. This process is summarized in Fig. 1.

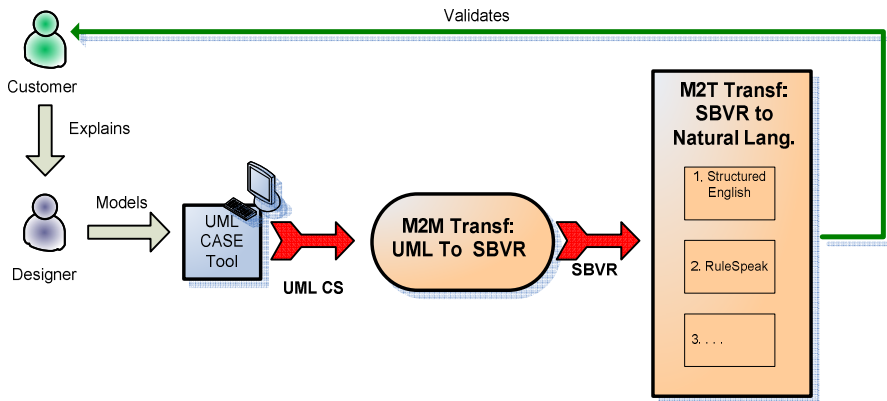


Fig. 1. Overview of the development process using our tool

## 2 Tool architecture

Our tool has been developed on top of the Eclipse IDE. The tool extends and combines existing Eclipse plug-ins to allow designers to model the UML CS, generate the corresponding SBVR instantiation and finally obtain a readable natural language (in Structured English) description of the CS. The full details of this tool support (including all our transformation specifications, examples and download instructions) can be found in [2].

As a first step, the initial UML schema is modeled using the UML2 Eclipse plug-in (an EMF-based implementation of the UML metamodel). Then, designers can execute our developed model-to-model transformations (implemented in ATL [6]) on the UML model to generate the corresponding SBVR representation as an instance of the SBVR metamodel. As SBVR metamodel we have used the internal metamodel developed as part of the MDT/SBVR component [7], optimized for development of tools that create and use SBVR vocabularies and rules. This metamodel is still under development so we have slightly extended it to be able to provide a full translation for all UML elements in the input model. The SBVR specification obtained as output of this transformation can be stored in an XMI file and imported into other SBVR tools.

As a next step, designers can translate this SBVR representation into a Structured English [3] natural language description. In particular, we have implemented a model-to-text transformation in MOFScript [8] that generates an HTML output that follows the notation (e.g. in terms of colors) used in the SBVR standard.

### 3 Conclusions and further work

We believe this tool is a first step towards a tighter integration of the business and conceptual modeling communities. Both share the same goals, such as the precise definition of concepts and business rules, but from different perspectives (the business one and the software modeling one). This integration can help to develop better systems by ensuring that the modeled system correctly represents the stakeholders' view and knowledge of the domain.

We are currently extending the tool in several directions. First, we are adding new ATL patterns to cope with the integrity constraints that may appear in the UML CS (either graphical ones, as *xor* or *subset*, or textual constraints expressed in OCL). These patterns have already been defined in [4] but not yet implemented due to the limitations of the current SBVR metamodel implemented in the Eclipse MDT/SBVR proposal [7]. Therefore, we have also aligned our efforts with the current Eclipse MDT/SBVR team working on its development to include all SBVR metaclasses required in our approach. Besides this, we plan to provide additional model-to-text transformations that allow generating natural language descriptions of the CS in different languages (e.g. Spanish) and styles (e.g. by adding support for the RuleSpeak English style).

### 4 Acknowledgments

This work was partly supported by the Ministerio de Ciencia e Innovación under TIN2008-00444 project.

### 5 References

1. I. Sommerville, P. Sawyer, Requirements engineering: a good practice guide. John Wiley & Sons, Inc. New York, NY, USA. 1997
2. J. Cabot, R. Pau, R. Raventós, UML-to-SBVR and SBVR-to-HTML transformations. <http://jordicabot.com/research/SBVR/index.html>
3. OMG: Semantics of Business Vocabulary and Rules (SBVR) Specification, v. 1.0 formal/08-01-02), 2008
4. J. Cabot, R. Pau, R. Raventós, From UML/OCL to SBVR Specifications: a Challenging Transformation. Information Systems Elsevier Journal, to appear. <http://dx.doi.org/10.1016/j.is.2008.12.002>
5. Eclipse Modeling Framework. <http://www.eclipse.org/emf>
6. ATL Project <http://www.eclipse.org/m2m/atl/>
7. MDT/SBVR Proposal <http://wiki.eclipse.org/MDT-SBVR-Proposal>
8. MOFScript Project <http://www.eclipse.org/gmt/mofscript/>