

DESIGN AND DEVELOPMENT OF BUSINESS RULES MANAGEMENT SYSTEM (BRMS) USING ATLAND ECLIPSE SIRIUS

Lynda Ait Oubelli and Mourad Oussalah

Department of Computer Science, University of Nantes,
2 Rue de la Houssinière BP-92208, 44322, Nantes, France

ABSTRACT

Nowadays, in the world of industry end-users of business rules inside huge or small companies claims that it's so hard to understand the rules either because they are hand written by a specific structural or procedural languages used only inside their organizations or because they require a certain understanding of the back-end process. As a result, a high need for a better management system that is easy to use, easy to maintain during the evolution process has increased. In this paper, the emphasis is put on building a business rule management system (BRMS) as a graphical editor for editing the models in a flexible agile manner with the assistant of ATL and Sirius frameworks within Eclipse platform. Thus, the proposed solution, on one hand, solves the problem of wasting resources dedicated for updating the rules and on the other hand it guarantees a great visibility and reusability of the rules.

KEYWORDS

MDE,DSM,BRMS,ATL transformation, EMF/ GMF,GraphicalEditor&Eclipse Sirius framework

1. INTRODUCTION

In the world of industry, the traditional approach that concerns the development of systems inside companies, based on code limits the flexibility of business services. Indeed, once the system is delivered, when the business logic changes, it will be necessary to make what we call an adaptation process by one or many stakeholders. In this context, the business rules have emerged. Which are senior formulations that describe the concepts of business and aim to separate the business logic from the system application logic, allowing the company to adapt its operations, according to dynamic market trends and conditions, to individual customer requirements or to the regulatory environment changes. Thus, we see the importance of how to represent, write and organize business rules to ensure on the one hand their better management and on the other hand, to facilitate their evolutionary maintenance. Hence, the need for a medium to better structure and managing the rules has increased. In this context lies our project entitled "Design and Development of Business Rules Management System (BRMS) Using ATL and Eclipse Sirius". It is worth noting that business rules represent the heart of each software package in a company. However, it is difficult to identify and to structure them for better management, readability, and reusability. For this, most of the companies always seek to create and develop tools to improve the functioning of their software packages. Actually, to modify a rule the end user should have strong computer science knowledge, Thereby, if a user of the standard solution wants to make a change in the logic of the software package, he must return to companies' personal staff to do that for him. But, this process is not practical for both companies and for the client because it

represents a loss of time and resources. Besides that the current structure of rules does not allow a good management. Thus a new modelling and structuring solution is highly required.

The remainder of this paper is organized as follows: Section 1 represents the general context and the problematic of our work, Section 2 describes an overview of the general context of the project; in this phase, we take a closer look at the theoretical and technical background. We establish, first, the basic concepts related to business rules. Secondly, we study the rules used in the module on which we will work as well as the language used in writing them. We choose later the different technologies and the tools that can be used to attend the expected goals. Next, in Section 3 we represent the proposed solution based on the theoretical sound of modelling, meta modelling and model transformation. Section 4 represents a case study of the proposal. Where we design, first, the new object meta model, we then model the business rules and we have finally developed the various ATL transformation rules that allow automating the transformation process of business rules from their current format to object format. Then, we develop an effective tool for managing these business rules. After that, Section 5 represents a discussion about well-known BRMS software's. Finally, Section 6 provides conclusions and future work.

2. BACKGROUND

2.1 MODEL TRANSFORMATION IN MDE

The preliminary study is an essential step in every project, in this section, we will introduce the basic concepts related to our project. We begin with a presentation of the general concepts of model driven Engineering; where [1] defined it as a discipline in software engineering that relies on models as first class entities and that aims to develop, maintain and evolve software by performing model transformations. According to [2] MDE embraces many other Technological spaces in a uniform way. An important aspect of MDE was presented by [3] which is, its emphasis on bridges between technological spaces and on the integration of bodies of knowledge developed by different research communities. In each area, the concepts of model, meta-model and model transformation take a different incarnation. But, what are the basic concepts of model-driven engineering? What is a model? What are the levels of modelling? What is a model transformation? What we mean by a DSM? According to [4] in MDE, models are considered as the unifying concept in computer science engineering. MDE promotes models to primary artefacts that drive the whole development process. The conceptual foundation of a modelling language, when expressed as a model, is called meta model. The relation between a model expressed in a language and the meta-model of this language is called Conforms To. Meta-models are in turn expressed in a modelling language called Meta modelling language. Its conceptual foundation is captured in a model called meta meta model, models, Meta models, and meta meta model form a three-level architecture with levels named M1, M2, and M3 respectively [5]. A standard meta meta model called Meta Object Facility (MOF) was proposed by [6]. An example of a meta model in the context of OMG standards is the UML meta model. It is worth noting that, model transformations in MDE follow a common pattern defined by [4] as model transformation pattern. The transformations are at the heart of the MDA approach. They allow getting different views of a model, to refine or abstract, they allow more to go from one language to another. It ensures the passage of one or more models of a given level of abstraction to one or more other models of the same level (horizontal processing) or a different level (vertical processing). From a source model to a target model, we need to identify the correspondence between the concepts of source and target models at their meta models.

The transformation of the source model is applied to automatically generate the target model by a program called transformation engine whose transformation rules are meant to perform the

transformation. The transformation model not only refers to the passage of M1 source to M1 target: we can find M1 to M2 (**promotion**), M2 to M1 (**downgrade**), M3 towards M1, M3 to M2... etc.

2.2. DSM GRAPHICAL EDITOR IN MDE

In recent years, a lot of efforts have been made by [7], [8], [9] and many others in order to provide facilities to design executable Domain-Specific Modeling Languages (xDSMLs). According to [10] creating a DSL in the MDE context is often called Domain-Specific Modeling (DSM). Building a graphical tool for DSM is often a very complex labor process. It needs a lot of time for developing and it often highly depends on reliable frameworks, because building an editor from scratch is mostly impossible nowadays. For example [11] described with high accuracy the Sirius framework used for developing a DSM Graphical Editor. Then [12] made a comparative study between two Frameworks Graphiti and Eclipse Sirius, they found that a Sirius based editor provides the most required features, which means that Sirius is a better option for creating a DSM editor. Which encourage us later to choose it to resolve our issues. Various meta tools that allow specifying editors were described by [13]. GMF [14] Xtext [15] and Sirius [16] support high-level specification of graphical and textual model editors. ATL [17], [18] supports the specification of model translations, EMFCompare [19] supports model differencing during the comparison process. Another challenge of software evolution faces MDE developers is studied by [20] which assume that the DSL itself is subject to evolution.

3. PROPOSED APPROACH

The present section is devoted to the presentation of the work done throughout the project using some printed screenshot to detail the implemented features. In this part, we will represent the workflow of the proposed solution and our new "Target Ecore meta model" of the used descriptive the language of the business rules.

3.1. MOTIVATION

Currently, our business rules as they are written follow a functional approach, in other words, they are composed of a series of instructions and procedures. This approach has the major disadvantage which is the complexity of maintenance during the evolution of the application. A simple change has a huge impact on other functions in the same business rule specifically and the application will be edited in its entirety generally. Moreover, the current representation of business rules does not allow better visibility of the structure and the content of the rule.

In the source meta model, we tried to model the language used in writing the rules and we finally realized a meta model that describes the structure of the business rules within a huge software package. After having reviewed this meta model, we realized that it does not offer a better modularity, scalability, and thereafter extensibility of the rules. For this reason, we thought to design once again the "Target Ecore meta model" of the concerned business rules. Inspiring by the advantages offered by the object-oriented approach.

3.2. THE SOURCE ECORE META MODEL OF THE BUSINESS RULES

Figure 1 below, shows "Source Ecore meta model ". As we mentioned before. It allows to well understanding the structure of a business rule. It was modelled by unified modelling language (UML).

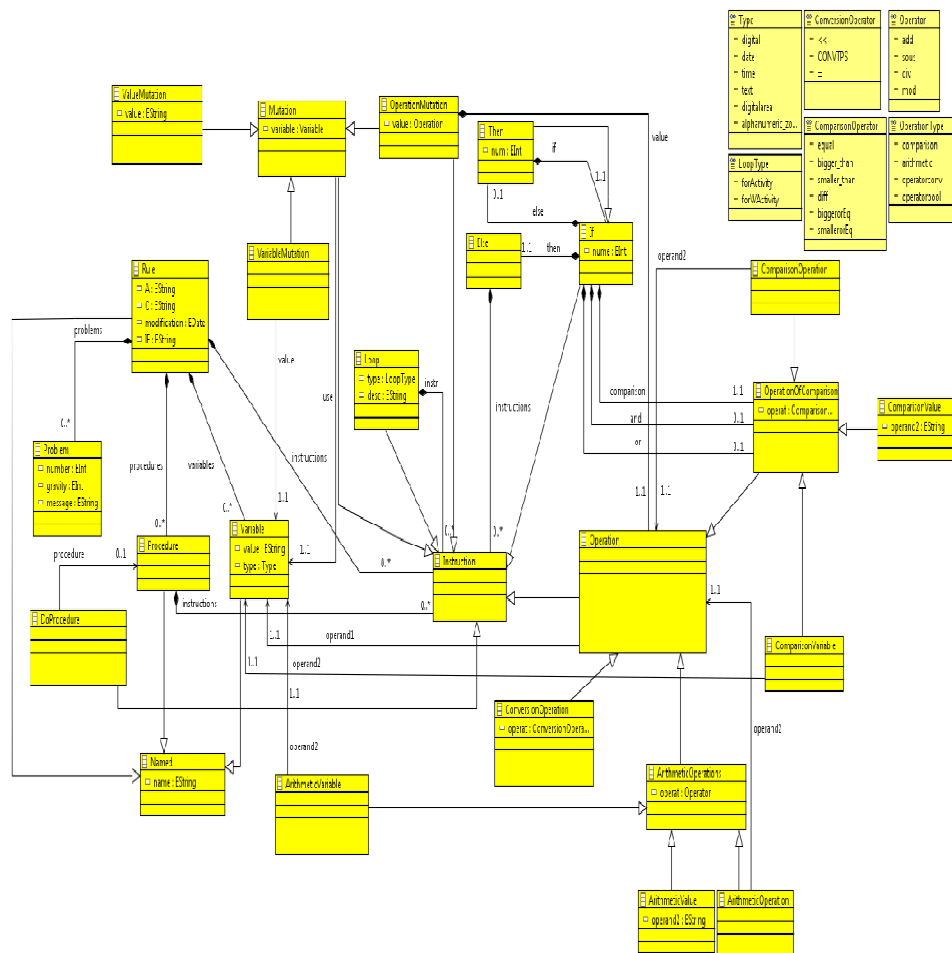


Figure 2. The Target Ecore metamodel of Business Rules

3.4. MODEL TRANSFORMATIONS USING ATL

To make a model transformation, ATL must take three inputs which are: source Ecore meta model, target Ecore meta model and a source xmi model which represents a rule that Conforms To source Ecore meta model in order to generate an xmi model that conforms to object Ecore meta model as it shows Figure 3.

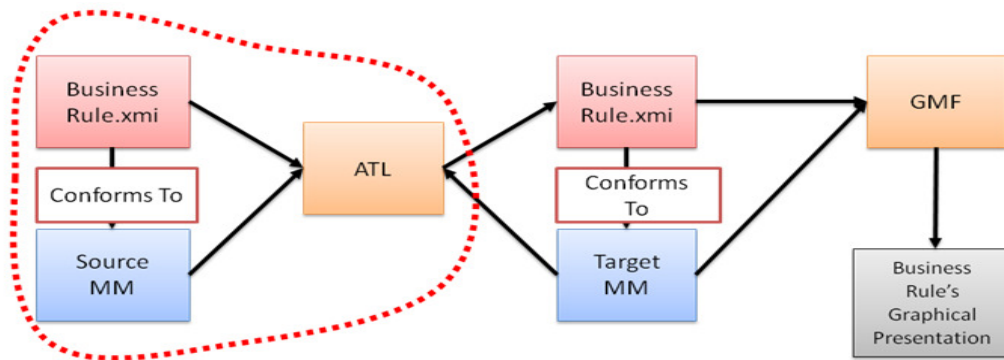


Figure 3. The workflow of the proposed solution for building the BRMS

We also write the transformation rules that tell ATL how to pass from an element of the source model to an element of the target model. As it shows Listing 1.

```

-- @path model_in=/BRMS/MetaModels/Source_MetaModel.ecore
-- @path model_out=/BRMS/MetaModels/Target_MetaModel.ecore

module AbstractionlevelsTransition2;
create OUT : model_out from IN : model_in;

rule Rule2Rule {
  from in: model_in!Rule
  to out: model_out!Rule (
    A <- in.Description.A,
    C <- in.Description.C,
    IE <- in.Description.IE,
    modification <- in.Description.modification,
    name <- in.Description.name,
    problems <- in.Problem,
    instructions <- in.Instruction,
    procedures <- in.Procedure
  )
}
  
```

Listing 1. Abstraction levels Transition Rules.atl between the Source and Target meta models

4. CASE STUDY

In what follows, we present a transformation example of a business rule that belongs to the software package. The latter contains a huge number of business rules that concern a specific domain, the used business rule during ATL transformation is presented as an .xmi file which conforms to source Ecore meta model.

4.1. EXAMPLE OF THE TRANSFORMATION PATH OF THE BUSINESS RULE

- *Step One:* First, we must introduce the two meta models source and target as .ecore files in the Eclipse modelling tools.
- *Step Two:* We instantiate, as an .xmi file, the business rule that we want to transform which Conforms To the source meta model.
- *Step Three:* We make the matching between the two meta models source and target by writing the ATL transformation rules.

Figure 4 shows the activity diagram of a business rule that we have used to validate our approach during the development of the BRMS, the concept of the rule is; in order to make loan amount from a bank, first, the bank evaluates person's credit score if it is less than 650\$ then, the bank evaluates person's employment history, if it is unstable then the bank evaluates person's misc loans amount, if it is heigh then it sets person's like hood of defaulting on a loan to high, else if person's credit score is ≥ 650 \$ or if person's employment history is stable or if person's misc loans amount is low then the bank set person's like hood of defaulting on a loan to ?

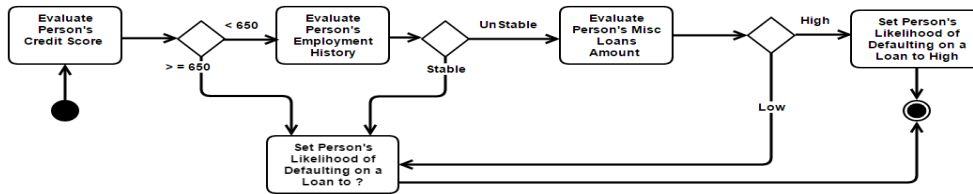


Figure. 4. UML activity diagram of the Business Rule

Figure 5 represents the xmi model of our business rule which conforms to the source Ecore meta model. Noting that:

- EPCS: Evaluate person's credit score,
- EPEH: Evaluate person's employment history,
- EPMLA: Evaluate person's Misc Loans Amount,
- PLHDLH: Set person's like the hood of defaulting on a loan to high.
- PLHDL: Set person's like the hood of defaulting on a loan to ?

After passing through these steps, we can finally generate the second model of the same business rule which conforms to the target meta model as it shows Figure 6. We draw the diagram manually, to make it clearer to the reader because the generated rule using ATL has the textual.xmi format.

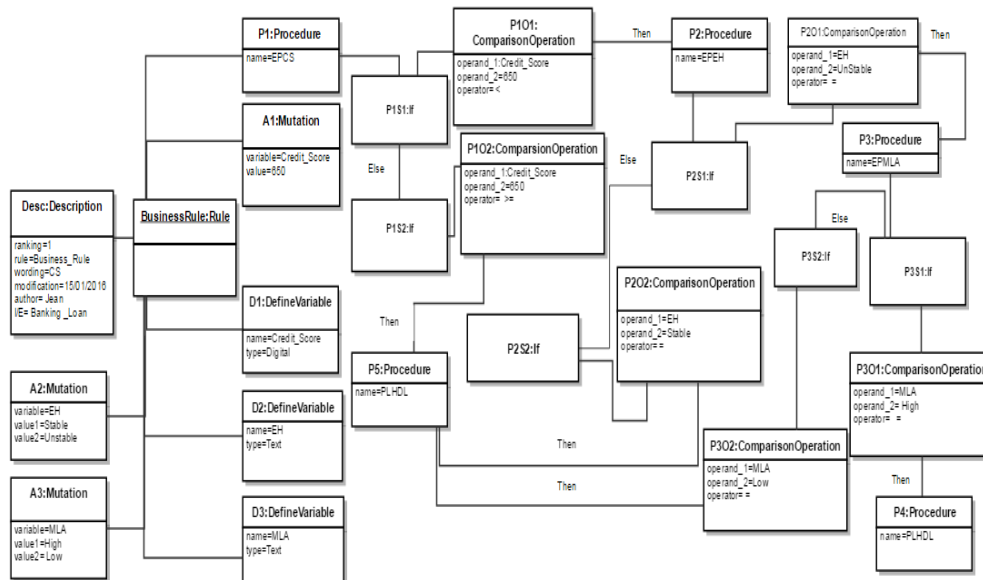


Fig. 5. The Business Rule model conforms To the Source Ecore meta model

4.2. LIMITS

One of the best methods to improve readability and structure of complex business rules is their graphical modelling. Another advantage offered by the graphical representation, is that it is very easy to understand by any type of end-users. It means, it didn't require any knowledge in computer science to use or to update it. While the .xmi format is understandable only by computer scientists. Unfortunately, the rules we have generated are under the .xmi format and we didn't find any tools to transform them in graphical format. For this, we find another way to solve this problem as it describes the next subsection.

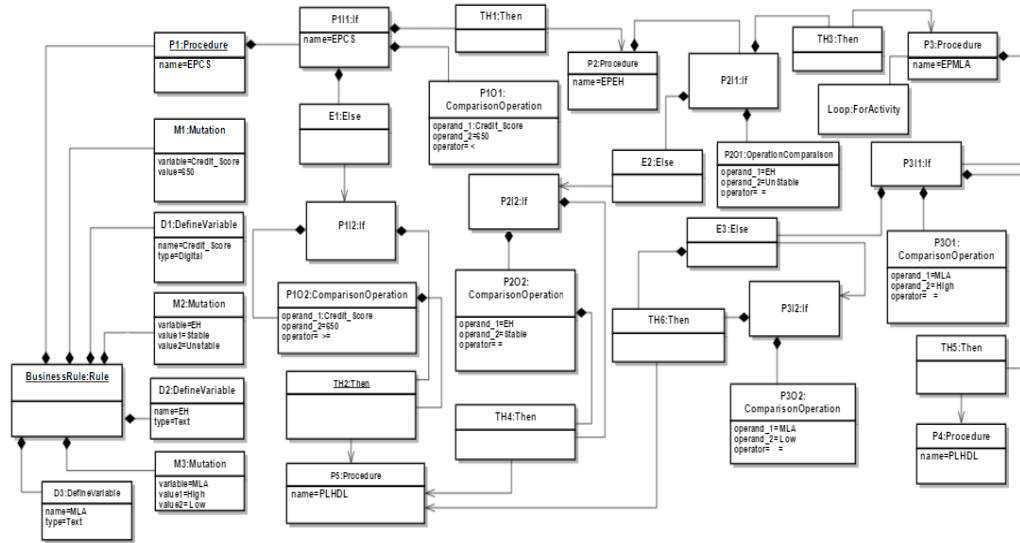


Figure 6. The Business Rule model conforms to the target Ecore meta model

4.3. GRAPHICAL MODELLING OF THE BUSINESS RULES

The graphical modelling of rules is extremely important in our project, that's why; we decided to develop a graphical editor that allows creating our BRMS as it shows Figure 3. This will greatly facilitate the work of end-users that are responsible for the update process. Between the disadvantages of GMF and the strength of Eclipse Sirius, we decided to use a rapid, agile, smooth path to develop the graphical editor as it shows Figure 7. From the business Rule that *Conforms To* the Target MM we can obtain a modelling tool that helps us to make a graphical presentation of the proposed business rule.

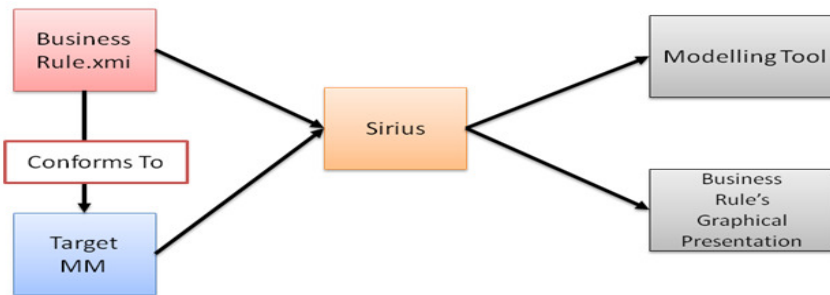


Figure 7. Presentation of Eclipse Sirius initiative for developing the BRMS

In this sub-section, we will detail the steps for creating the editor with Sirius, explaining the associated key concepts.

- *Create the EMF model:*The step of creating the EMF model is already made in the previous section. Our DSM (Domain Specific Model), is a language dedicated to a business domain with its keywords and its formalism, it was described by the meta model already presented "Target Ecore meta model ", it specifies the structure of a business rule and relationships with the various components of this business rule. In this step an .ecore file is realized.
- *Creation of GenModel:*Once the ".genmodel" file created, then, by opening it and start the generation of the three plugins, "*target.model*", "*target.model.edit*"and "*target.model.editor*". Next, by starting the Eclipse IDE with the generated plugins we open the "Run Configurations" panel via the menu " Run > Run configurations ", and then we create a new configuration of the type "Eclipse Application".
- *Creating the editor:*The great strength of Sirius is, it allows the specification of editors in a completely graphic manner, without knowing anything about back-end process. Parallel to the specification, Sirius allows things to be observed in real time starting from instantiating the model that we want to represent.

We will put in this section the Sirius pallet that we have created. The latter can be used to create two types of elements: in one hand the visible elements in the pallet and on the other hand the mechanisms to change the items of each businessrule on the fly. For clarity and for easy handling, we have grouped the elements in subsections. We can distinguish in Figure 9 above three types of elements. Nodes that represent simple instructions, defining problem and variable declaration. A container which represents items that can contain sub-elements as the "if" conditional statement or the loop "for each". And the third type of elements (which will not be visible to the end user of the pallet) that allows making changes on the nodes and containers without going through the pallet such as: adding new element, deleting an element, renaming element, fixing errors in the model, changing the label of an element, relationships source and target And even adding Java services for *arithmetical computations on tables*. The figure aboveshow the creation of the business rule with the modeler, it highlights the three views with which the user has the possibility to create, read and update it.

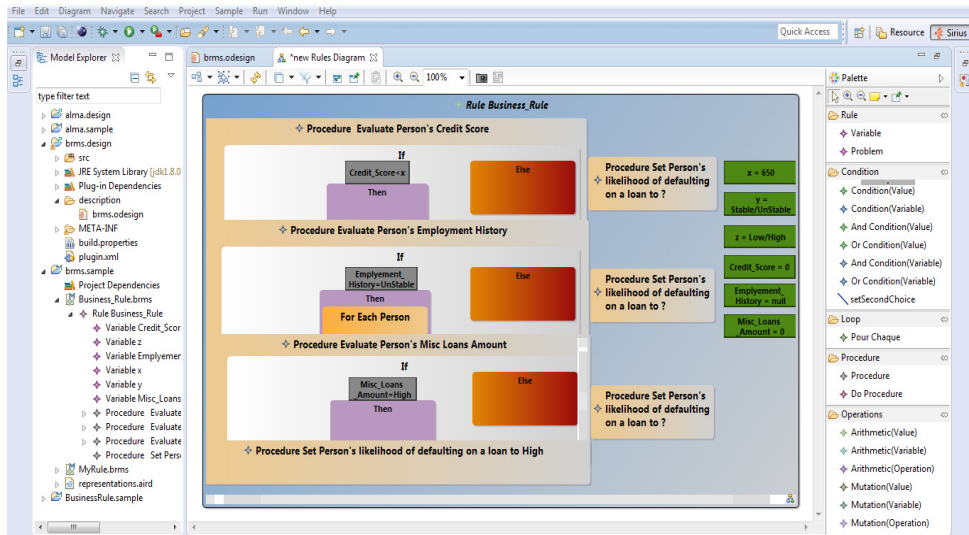


Figure 8. Domain-specific IDE for model-driven development of the Business Rule

5. DISCUSSION

Business Rules Management Systems are the softwares that can create, manage and support the business rules of an organization or a company. Thus, we assumed that their use will accelerate the creation of the business rules for software packages. Our research has shown that there are multiple softwares of BRMS already used by companies. However, none fully meets the objectives of the project. The following table 1 provides a comparison of the top five BRMS softwares that we thought, it would be possible to adapt them to our project.

	Jess	DTRules	Drools	M.D Network	ILOG	Proposed Solution
Gratuitousness		★★★★		★★		★★★★
Popularity	★★	★	★★★★	★★	★★★★	★★★★
Portability	★	★★	★★★★	★★★★	★★★★	★★★★
Intuitiveness	★	★★	★★	★	★	★
Adaptability	★	★	★★	★★★★	★★★★	★★★★
Scalability	★	★★	★★★★	★★★★	★★★★	★★★★

Table 1. Classification Table of top BRMS software.

This table was completed following the results of test cases made immediately on software and from opinions gathered by other users. According to our research, the most powerful softwares

are ILOG (IBM) and Microsoft Developer Network, but their use is not free and requires in addition to be trained and have a solid foundation in computer science. Therefore, they do not meet the expectations of the project. The other three programs (Jess, Drools and DTRules) are much less performing and require from their users much knowledge in computer science and a lot of time to control and adapt the tool to their needs.

Thus, these softwares do not meet the expectations of the project. However, the results proved that our proposed solution for the BRMS covered all features and simplifies the product specification, reduces design time and rapidly increases the overall productivity without any knowledge about the back-end processes.

6. CONCLUSION AND FUTURE WORK

This technical paper has described the steps that we have faced during the study of the project. It details the stages through which we passed to achieve the project objectives. We started with a general presentation of the project followed by a theoretical and technical study. Then we explained the work that we have done during the project. Finally, we finished by the proposed perspectives. This project was a considerable use. Indeed, it allowed exploiting the theoretical and technical knowledge, such as ATL for model transformation, the project Eclipse Sirius for creating graphical editors. Thus, the complexity of the rules was a great challenge that pushes us to find an easy way for the end users to manipulate, update them without any difficulties.

After our research studies, design and development processes, we succeeded to build the BRMS as a very flexible graphical editor. We propose as perspectives, the generation in text format rules that are consistent with target ecore meta model. Indeed, after having modelled the business rules graphically using the editor that we have developed, we can obtain as an output XMI file. The latter may be converted to text format through XSLT which is a Document's transformation tool defined in the recommendation XSL of W3C as shows Figure 9. It allows the transformation of an XML document into another format, as PDF or HTML. Thus, the rule will be ready to use in text format.

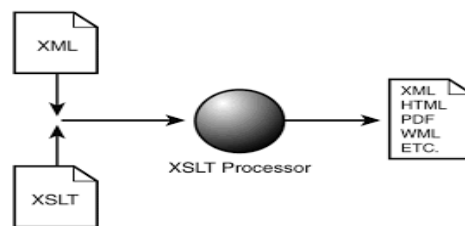


Figure 9. XSLT Processing

Actually, we have a graphical representation of the business rules within their software package, we can do the opposite path from the one described in the previous paragraph, i.e. from text format, we can obtain an XML file that can be represented graphically, this treatment can be automated subsequently to transform all the existing XML rules.

REFERENCES

- [1] KURTEV, Ivan & VAN DEN BERG, Klaas, (2003) "A synthesis-based approach to transformations in an MDA software development process". In: Model Driven Architecture: Foundations and Applications, p. 121.

- [2] VARRÓ, Dániel & BALOGH, András, (2007) “The model transformation language of the VIATRA2 framework. *Science of Computer Programming*”, vol. 68, no 3, p. 214-234.
- [3] KLEPPE, Anneke G., WARMER, Jos B., & BAST, Wim, (2003) “MDA explained: the model driven architecture: practice and promise”, Addison-Wesley Professional.
- [4] JOUAULT, Frédéric, ALLILAIRE, Freddy, BÉZIVIN, Jean, & al, (2008) “ATL: A model transformation tool. *Science of computer programming*”, vol. 72, no 1, p. 31-39.
- [5] JOUAULT, Frédéric, BÉZIVIN, Jean, & KURTEV, Ivan, (2006) “TCS: a DSL for the specification of textual concrete syntaxes in model engineering”. In :Proceedings of the 5th international conference on Generative programming and component engineering. ACM. p. 249-254.
- [6] OMG/MOF meta object facility (MOF) specification, OMG Document AD/97-08-14, September (1997). Available from: www.omg.org.
- [7] COMBEMALE, Benoît, CRÉGUT, Xavier, & PANTEL, Marc, (2012) “A Design Pattern to Build Executable DSMLs and Associated V&V Tools”, in: APSEC, IEEE. Vol. 1, pp. 282-287.
- [8] MAYERHOFER, Tanja, LANGER, Philip, WIMMER, Manuel, & al, (2013) “ xMOF: Executable DSMLs based on fUML”, in: SLE, LNCS 8225, Springer International Publishing, p. 56-75.
- [9] TATIBOUËT, Jérémie, CUCCURU, Arnaud, GÉRARD, Sébastien, & al., (2014) “Formalizing Execution Semantics of UML Profiles with fUML Models”, in: MODELS, LNCS 8767, Springer, pp. 133–148.
- [10] HUDAK, Paul, (1998) “Modular domain specific languages and tools. In *Software Reuse*”, Proceeding of IEEE Fifth International Conference, pp. 134-142.
- [11] VIYOVIĆ, Vladimir, MAKSIMOVIĆ, Mirjam, & PERISIĆ, Branko, (2014) “Sirius: A rapid development of DSM graphical editor”, in IEEE 18th International Conference on Intelligent Engineering Systems INES, pp. 233-238.
- [12] VUJOVIĆ, Vladimir, MAKSIMOVIĆ, Mirjana, & PERIŠIĆ Branko. , (2014) “Comparative analysis of DSM Graphical Editor frameworks: Graphiti vs. Sirius”, 23rd International Electrotechnical and Computer Science Conference ERK, Portorož, Slovenia, pp. 7-10.
- [13] VAUPEL, Steffen, STRÜBER, Daniel, RIEGER, Felix, & al., (2015) “Agile bottom-up development of domain-specific IDEs for model-driven development”, in Proceedings of FlexMDE, pp.12-21.
- [14] GMF Tooling, (2014) “Graphical Modeling Framework”. Available at: goo.gl/9Lgtzf
- [15] BETTINI, Lorenzo, (2013) “Implementing Domain-Specific Languages with Xtext and Xtend. Packt”, Publishing Ltd.
- [16] Eclipse Sirius, (2016) “The easiest way to get your own Modeling Tool”. Available at: goo.gl/mRBHQG
- [17] Lina, (2008) “Atlas Transformation Language”. Available at: [Http: //www.sciences.univ-nantes.fr/lina/atll/](http://www.sciences.univ-nantes.fr/lina/atll/).
- [18] Eclipse ATL, (2016). Available at: goo.gl/ubISKw
- [19] Eclipse Foundation, (2011) “Emf compare, A developer guide”, available at: <http://goo.gl/53XASj>.
- [20] FEHRENBACH, Stefan, ERDWEG, Sebastian, & OSTERMANN, Klaus, (2013) “Software evolution to domain-specific languages”, Springer International Publishing. In International Conference on Software Language Engineering, pp. 96-116.

AUTHORS

Lynda AIT OUBELLI is currently a PhD student in Computer Science. Previously she was a Graduate Research Student of Software Architecture in Nantes University, France; she received her bachelor's and Master's degrees in Software Engineering from Souk-Ahras University, Algeria. Her fields of research interests include MDE {Modeling, Meta Modeling, Model Transformation, Model Evolution, Model co evolution, Model comparison} Object Oriented Approach and Formal modelling languages. She can be reached at (lynda.aitoubelli@univ-nantes.fr).

Mourad Chabane OUSSALAH is a Professor of Computer Science at the University of Nantes, France and the chief of the software architecture modelling Team. His research concerns software architecture, object architecture and their evolution. He worked on several European projects (Esprit, Ist,...). He is (and was) the leader of national project (France Telecom, Bouygues telecom, Aker-Yard-STX...)..He earned a BS degree in Mathematics in 1983, and Habilitation thesis from the University of Montpellier in 1992.He is also the co-founder of the National RPO, OCM (Objects, Components and Models) and LMO (Langages et Modeles a objets) Conferences since 1992, and the founder of the CAL conference (Conference

francophone sur les architectures logicielles) and the general chairman of the EWSA (European Workshop on Software Architecture) 2006.

Prof. Oussalah has published 4 books and over 150 refereed journal and conference papers in the fields of Software Architecture, Software Engineering, and Object Oriented Modelling. He is listed among the most prolific authors in the DBLP computer science bibliography. He has served on Program Committees of c.a. 50 International Conferences and Workshops and has act as referee for 10 International Journals.

His contact address is: LINA Laboratory, CNRS UMR 6241, University of Nantes, 2, Rue de la Houssinière, BP 92208, 44322 Nantes, France . (Mourad.Oussalah@Univ-Nantes.fr).