

A MODEL DRIVEN ARCHITECTURE TOOL BASED ON SEMANTIC ANALYSIS METHOD

Thiago Medeiros dos Santos, Rodrigo Bonacin

CESET, Unicamp and CenPRA, MCT, Rua Paschoal Marmo, 1888, Limeira, SP, Brazil

M. Cecilia C. Baranauskas

Institute of Computing, Unicamp, Caixa Postal 6176, 13083 970, Campinas, São Paulo, Brazil

Marcos Antônio Rodrigues

CenPRA, MCT, Rodovia Dom Pedro I, km 143,6, 13069-901, Campinas, São Paulo, Brazil

Keywords: Organizational Semiotics, Model Driven Architecture, Model Driven Engineering.

Abstract: Literature in Information Systems presents a set of approaches to analyzing and modeling the organizational context in order to improve the quality of the computational systems. The research community also looks for alternatives for aligning these approaches with Software Engineering techniques, which support the large scale and low cost software production and maintenance. In this work is proposed an approach and a tool to support the construction of computational models taken from the organizational models. This approach is based on concepts, methods and techniques from MDA (Model Driven Architecture), and from Organizational Semiotics, more specifically the MEASUR (Methods for Eliciting, Analyzing and Specifying User's Requirements). A MDA tool was constructed in order to realize the approach. The first version of this tool provides means to model Ontology Charts. By using semi-automated transformations, it is also possible to produce UML (Unified Modeling Language) class diagrams from the modeled Ontology Charts. The tool features, design solutions, and capabilities are explained.

1 INTRODUCTION

The design and use of computational systems are susceptible to the organizational context where they are immersed. Literature presents a set of approaches to analyze and to model the organizational context in order to improve the quality of computational systems.

The research community also looks for alternatives for aligning these approaches with Software Engineering techniques, which support the large scale and low cost software production, and maintenance. Among the main difficulties is the multidisciplinary nature of the problem, which require methods, techniques and tools coming from many domains, including: humanities, to understand and model the organizational context; engineering, to deal with aspects related to large scale and low cost production; and technologies, to construct appropriate techniques and tools.

In this work we propose an approach and tool to support the construction of computational models derived from the organizational models. The main objectives of the proposed approach are to allow the representation of the organizational context semantics, and to provide a way to quickly produce corresponding design models.

In order to achieve these objectives, we make use of concepts, tools, and methods from Model Driven Architecture (MDA) (OMG, 2003) and Organizational Semiotics (OS) (Liu, 2000). OS research explores the use of signs and their effects on social practices. OS understands that each organized behavior is affected by the communication and interpretation of signs by people, individually or in groups. This work is based on Stamper's school of OS (Stamper et al., 1988; Stamper 2001), more specifically the use of the MEASUR (Methods for Eliciting, Analyzing and Specifying User's Requirements) (Stamper et al.,

1988) to model the semantic aspects of the organization.

The MDA is a proposal that permits multiples refinement of abstract models to generate the systems code, through transformations (Gardner et al., 2003). This work explores how Computation Independent Models (CIM) can be refined using the MDA approach.

A MDA tool was constructed in order to materialize the approach. The first version of this tool supports the modeling of Ontology Charts produced during the Semantic Analysis Method (from MEASUR). By using semi-automated transformations, it is also possible to produce UML (Unified Modeling Language) class diagrams from the modeled Ontology Charts. In this paper, the tool features are explained through a small example of ontology chart.

The paper is organised as follows: Section 2 presents the Model Driven Architecture and the Semantic Analysis Method, Section 3 discusses the proposed approach to produce Platform Independent Models from Computation Independent Models, Section 4 presents Sonar, the developed tool, Section 5 discusses the approach and further works, and Session 6 concludes.

2 BACKGROUND

On this section we present concepts and methods from Model Driven Architecture (MDA) and the Semantic Analysis Method (SAM).

2.1 Model Driven Architecture

MDA is an Object Management Group (OMG) proposal to promote an open solution for the challenges related to constant business and technological changes. To reach this objective in MDA the application logic and the technological issues of a specific platform are specified in models of different abstraction levels.

In order to introduce MDA, firstly is presented what the OMG means by the following concepts:

- *Architecture*: The OMG adopts the Shaw and Garlan (1996) concept. “The architecture of a system is a specification of the parts and connectors of the system and the rules for the interactions of the parts using the connectors.” (OMG, 2003, p.2-3) The MDA specifies kinds of models and their relations.
- *Platform*: “A platform is a set of subsystems and technologies that provide a coherent set of

functionality through interfaces and specified usage patterns, which any application supported by that platform can use without concern for the details of how the functionality provided by the platform is implemented.” (OMG, 2003, p.2-3)

- *Viewpoint*: “A viewpoint on a system is a technique for abstraction using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within that system.” (OMG, 2003, p.2-3)
- *Model*: “A model of a system is a description or specification of that system and its environment for some certain purpose.” (OMG, 2003, p.2-2)

In the MDA specification three types of viewpoints are proposed, as well as one model for each viewpoint: the CIM (Computation Independent Model), the PIM (Platform Independent Model) and the PSM (Platform Specific Model). The CIM, which is the focus of this work, has an important role as a bridge between developers and domain specialists. The CIM specifications are requirements for PIM and PSM.

These models are specified according to metamodels (models that describe models) that should be specified according to metametamodels. The metametamodels should be self-described. Other metamodel layers could also be created if necessary. According to MetaObjectFacility 2.0 (MOF) specification (OMG, 2006), the fundamental concepts can be used to handle any number of layers.

In MDA one model can be converted to another model of the same system (usually more specific) through a transformation process. The transformation follows mappings, which can be understood as a set of rules and methods to obtain a new model from a previous one. The Model Type Mappings allows specifying transformations from any model built using types specified in PIM to models expressed using PSM types. The Model Instance Mappings allows specifying model elements which should be transformed in a particular way.

According to Brown and Conallen (2005), some times it is not possible to obtain automatic transformations from model to model. The transformation rules must be clear and unambiguous, and should be able to programmatically access the necessary elements of the models. Brown and Conallen (2005) also emphasize that most of the steps that involve natural language documents reading are not typically suitable to MDA-style of automation.

This is the main reason why automated transformations from CIM to PIM are not common. Usually, CIM models are based on textual descriptions (i.e. Use Cases). However, better structured CIM models, based on a visual language, can help to construct systems aligned to business and human factors, and even the “semi-automated” transformations could bring benefits associated to the cost, time and quality of the software. The next section presents the Semantic Analysis Method as an example of a CIM model constructed according to Organizational Semiotics concepts.

2.2 Semantic Analysis Method

The Semantic Analysis Method (SAM) assists the users or problem owners in eliciting and representing their requirements in a formal and precise model. With the analyst in the role of facilitator, the required system functions are specified in an ontology chart, which describes a view of responsible agents in the focal business domain and their behavior or action patterns named affordances (Liu, 2000). Some basic concepts of SAM adopted in this paper are based in Liu (2000):

- “The world” is socially constructed by the actions of agents, on the basis of what is offered by the physical world itself;
- “Affordance”, the concept introduced by Gibson (1977), is used to express invariant repertoires of behavior of an organism made available by some combined structure of the organism and its environment. In SAM (Stamper, 1993) the concept introduced by Gibson was extended by Stamper to include invariants of behavior in the social world;
- “Agent” can be defined as something that has responsible behavior. An agent can be an individual person, a cultural group, a language community, a society, etc. (an employee, a department, an organization, etc.);
- “An ontological dependency” is formed when an affordance is possible only if certain other affordances are available. The affordance “A” is ontological dependent on the affordance “B” means that “A” is only possible when “B” is also possible;
- “Determiners” are properties which are invariants of quality and quantity that differentiate one instance from another;
- “Specialization”, agents and affordances can be placed in generic-specific structures according to whether or not they possess shared or different properties;

The concepts of Semantic Analysis are represented by means of ontology charts, which have a graphical notation to represent agents (circles), affordances (rectangles), ontological dependencies (lines drawn from left to right), role-names (parentheses) and whole-part relations (dot).

3 A PROPOSAL FOR BUILDING CIM TO PIM TRANSFORMATIONS

Ontology Chart can be understood as a CIM model given that it focuses on organization and requirements. However, the SAM addresses issues that are not represented in any of the UML diagrams and it provides a different way of thinking about the organization if compared with the Object Oriented paradigm. Consequently the transformation from Ontology Chart to Class diagrams is not a trivial task. It is also possible to see the Ontology Chart as a kind of CIM metamodel (not adopted in this work), which specifies the semantics of other models.

The rationale to construct the transformation rules is based on a previous work that deals with how to build UML Class Diagram informed by SAM (Bonacin et al., 2004). Nevertheless, additional aspects must be considered in CIM to PIM transformations. The following steps are proposed to build CIM to PIM transformations: (1) create the transformation rules, (2) develop the metamodels for CIM, (3) analyze which rules need human intervention, (4) create the heuristic metamodels, (5) design the interfaces to facilitate the human intervention, and (6) develop the transformations.

Regarding the rule creation, in this work was adopted a group of ten heuristic rules proposed in Bonacin et al. (2004). The next step was the development of meta-models for ontology charts. The KM3 (Kernel MetaMetaModel) language (Atlas, 2005) facilitated the description of Metamodels in MOF (*MetaObject Facility*) (OMG, 2006), and the Ecore (Budinsky et al.; 2003).

Figure 1 presents an UML diagram for the Ontology Chart Metamodel. The concept of “element” was created to describe anything at the diagram. Affordance is a central concept at the diagram. Role-name and agents are special types of affordances. Affordances may also include determiners and specializations. Ontological dependencies link affordances.

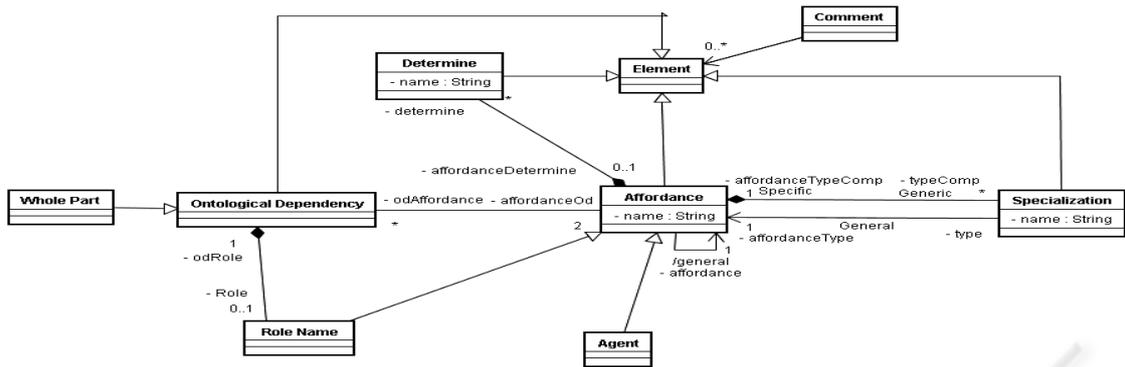


Figure 1: Ontology Chart Metamodel.

Whole-part is a special type of ontological dependency and an ontological dependency may also include a role-name.

After constructing the Metamodel, the proposal was to find out the heuristics that could be directly automated, and which rules need human intervention. An example of a non-directly automated transformation is the creation of classes or operations from affordances names. It is necessary a human intervention to say whether an automated rule should be applied or not, and to decide which automated rule should be applied: to transform the affordance into a class or operation.

If the second option is selected, additional information is necessary to construct the PIM. Figure 2 shows an intermediary model proposed to describe this information. It has the “HAffordance” Class with three attributes: *Operation* indicates if the affordance will be mapped to an operation or to a class, *ClassId* indicates the class that will contain the operation, and *affordanceId* identify the affordance that will be mapped.

The next step is the design of the interfaces for end-user intervention. A wizard based solution was adopted in this work. The last step is the implementation of the transformations. They map data from the ontology chart metamodel and the heuristic model to PIM (UML models).

transformation occurs internally in this modeling tool. This fragment has the “society” as the root agent; “person” and “thing” is ontologically dependent of “society”; and “owns” is ontologically dependent of “person” and “thing”.

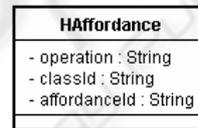


Figure 2: Example of Heuristic Metamodel.

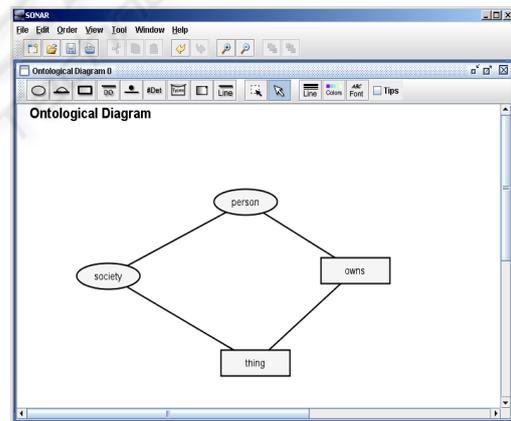


Figure 3: Sonar Main Window.

4 SONAR TOOL

As Figure 3 shows, the Ontology Chart modeling tool, named Sonar, has the basic features of a CASE (*Computer-Aided Software Engineering*) tool, including: icons to model the diagram elements, copy, paste, undo, redo, zoom, and export as image. The fragment of the ontology chart diagram in Figure 3 will be used to explain how the

After modeling the Ontology Chart, the user can transform it to UML models by using wizards on the “Tool” menu. Figure 4 shows an example of wizard where designers can specify which affordance will be mapped to operation, and associate them to classes (affordances mapped to classes) that will contain it. For example the class “Person” contains the method “owns”. By associating “owns” to “person” a heuristic model is created following the heuristic metamodel specification. The heuristic

model contains the necessary data to determine the appropriated transformations that will be applied.

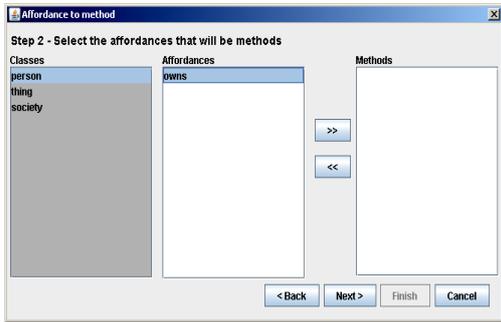


Figure 4: Example of wizard to human intervention.

Figure 5 shows an overview of the transformations implemented by the Sonar tool. At the CIM level there are two models: the ontology model constructed according to the diagram, and the heuristic model that is a result of human intervention through wizards. These models follow the metamodels specified in MOF and/or Ecore. Internally the models are represented according to the XMI format (OMG, 2007).

Transformations (Figure 5) were constructed using the ATL (*Atlas Transformation Language*) (Atlas, 2006). “ATL provides developers with a way to produce a number of target models from a set of source models.” (Atlas, 2006, p.1).

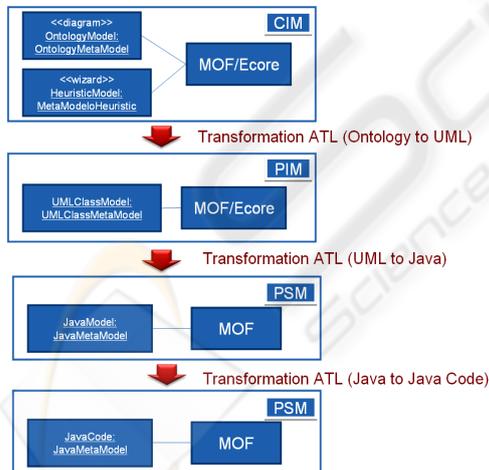


Figure 5: Sonar Transformations.

There are two source models (ontology and heuristic) that produce target models. After producing the PIM a number of transformations, can be applied to produce PSM models, such as: to C++, to EJB, and to SQL. According to Figure 5, in this version of Sonar we can transform the UML to a Java model and finally to Java code.

Figure 6 is a graphical representation of a possible UML model produced from the ontology model presented at Figure 3. This model contains three classes produced from the “Person”, “Thing” and “Society”. The user has decided to transform the “Owns” affordance in a method of the class “Person”. The method could return, for example, the thing(s) that the “Person” owns. Alternative models can be produced according to the human intervention, such as, one model without the “Society” class.

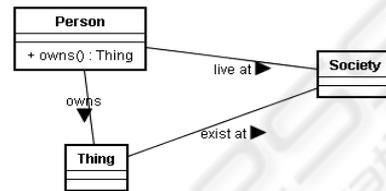


Figure 6: Constructed UML Model.

5 DISCUSSION AND FURTHER WORK

The first version of Sonar can be understood as a starting point to further research. One important feature of the proposed approach is the possibility of following the process: the capacity to detect the origin and follow the tracking from the source model to the produced model. Although the process includes human intervention, the decisions are registered on the heuristic models. Thus, it is possible to verify how elements of the PIMs were constructed from the CIM elements.

However, Ontology Charts do not produce a complete PIM; consequently the PSM is also incomplete. In a real life situation, the produced PIM should be complemented with many details before the transformation to PSM. In addition, important decisions need human intervention (i.e. if implicit elements of the CIM, like “society” should or not be explicit in the implemented system). Nevertheless, it is necessary to consider two main aspects:

- The produced model can be a valuable starting point for the design of a complete and consistent PIM. It produces a set of domain classes that are closely related to the real context. Usually design models closely related to real contexts result on systems more reusable and easier to maintain;
- The approach can also be applied to equally well define CIM (in addition to Ontology Chart), to produce a more complete PIM, e.g.: a “norm based model”, from the OS norms analysis

method (Stamper et al., 1988), could be used in order to model the dynamic aspects of the organization, consequently improving the produced PIM.

It is also important to consider the possibility to produce other diagrams from the Ontology Charts. Ontology has been used to deal with problems of systems integration. The Semantic Web (W3C, 2007) makes use of ontology to process the content of information. The Ontology Charts could be used to produce models based on technologies such as: OWL (Web Ontology Language) (W3C, 2004). As Semantic refers to meaning, and meaning are socially created by humans, is expect to create a more faithfulness Ontology if we start from a Computer Independent Model.

6 CONCLUSIONS

The design and use of computational systems are susceptible to the organizational context where they are immersed. In this paper is proposed an approach to produce computational models transformed from the organizational ones. This approach has its basis on techniques and methods from MDA and SO.

An Ontology Chart modeling tool, named Sonar, implements the main proposed ideas. It shows the possibility of generating PIM from CIM, using semi-automated transformations. Among the main benefits are the visibility of the transformation process, and the production of an initial design model closely related to the real context concepts.

ACKNOWLEDGEMENTS

We thank colleagues from CESET and IC at Unicamp, and CenPRA for valuable contributions.

REFERENCES

- Atlas group, 2005. *Kernel MetaMetaModel v.0.3*. Retrieved December, 13, 2007, from <http://www.eclipse.org/gmt/atl/doc/KernelMetaMetaModel%5Bv0.06%5D.pdf>
- Atlas group, 2006. *ATL User Manual.v. 0.7*. Retrieved December, 13, 2007, from http://www.eclipse.org/gmt/atl/doc/ATL_User_Manual%5Bv0.7%5D.pdf
- Bonacin, R., Baranauskas, M. C. C., and Liu, K., 2004. From Ontology Charts to Class Diagrams: semantic analysis aiding systems design. Proceedings of *6th International Conference on Enterprise Information Systems*, Porto, Portugal.
- Brown, A., and Conallen, J., 2005. *An introduction to Model Driven Architecture Part II: Lessons from the design and use of an MDA toolkit*. The Rational Edge. Retrieved December, 13, 2007, from <http://www.ibm.com/developerworks/rational/library/apr05/brown/index.html>
- Budinsky, F., Steinberg, D., Merks, Ed, Ellersick, R., Grose, T. J., 2003. *Eclipse Modeling Framework: A Developer's Guide*, Addison Wesley Professional.
- Gardner, T., Griffin, C., Koehler J., and Hause, R. *A review of OMG MOF 2.0 Query / Views / Transformations Submissions and Recommendations towards the final Standard*. IBM Zurich Research Laboratory, Retrieved December, 13, 2007, from <http://www.zurich.ibm.com/pdf/ebizz/gardner-et-al.pdf>
- Gibson, J.J, 1977. The Theory of Affordances. In *Perceiving, Acting, and Knowing*, Eds. Robert Shaw and John Bransford.
- Liu K. (2000). *Semiotics in information systems engineering*, Cambridge. University Press.
- Object Management Group (OMG), 2003. *MDA Guide Version 1.0.1*. Eds. J. Miller and J. Mukerji. Retrieved December, 13, 2007, from <http://www.omg.org/docs/omg/03-06-01.pdf>
- Object Management Group (OMG), 2006. *Meta Object Facility (MOF) Core Specification, Version 2.0*. Retrieved December, 13, 2007, from <http://www.omg.org/docs/formal/06-01-01.pdf>
- Object Management Group (OMG), 2007. *MOF 2.0/XMI Mapping, Version 2.1.1*. Retrieved December, 13, 2007, from <http://www.omg.org/docs/formal/07-12-01.pdf>
- Shaw, M. and Garlan, D., 1996. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall.
- Stamper, R. K., Althaus, K. e Backhouse, J., 1988. MEASUR: Method for Eliciting, Analyzing and Specifying User Requirements. In Olle, T. W., Verrijn-Stuart, A. A. e Bhabuts, L. (eds.) *Computerized Assistance During the Information Systems Life Cycle*. Elsevier Science, Amsterdam, 67-116.
- Stamper, R., 1993. Social Norms in Requirements Analysis – an outline of MEASUR. In *Requirements Engineering, Technical and Social Aspects*, Eds. M. Jirotko, J. Goguen, and M. Bickerton, Academic Press, New York.
- Stamper, R. K., 2001. Organisational Semiotics: Informatics without the Computer? In *Information, Organisation and Technology: Studies in Organisational Semiotics*, eds. K. Liu, R. Clarke, P. B. Andersen and R. K. Stamper. Kluwer Academic Publishers.
- W3C, 2004, *OWL Web Ontology Language Reference*, Retrieved December, 13, 2007, from <http://www.w3.org/TR/owl-ref/>
- W3C, 2007, *Semantic Web Activity*, Retrieved December, 13, 2007, from <http://www.w3.org/2001/sw/>