IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4734

Describing W2000 Using RDF When the Meaning Is Essential

Luca Contursi, Roberto Paiano
Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce
Via per Arnesano, 73100, Lecce, Italy
Tel{+39} 0832 320229, Fax:{+39} 0832 320279
{luca.contursi, roberto.paiano}@unile.it

ABSTRACT

A powerful methodology for the conceptual modeling of hypermedia and Web applications is a must for designers. In addition to the advantages of easily and fully expressing an idea and of discussing it with the customer, to better understand what he needs, a conceptual design helps the designers managing the complexity and letting them completely describe how the application will be and how it will appear to different users and on different devices. Every schema they produce will be precious for people who will develop the application, using any language or in any environment, for anyone who will work on it. But, in order to get the best result and to be sure that every effort will give what the designers expect, we must be sure that the model is fully understood by everyone. The meaning of each schema must be absolutely clear. This is even a bigger problem when the output of the modeling activity is used to feed a prototyping system or a development framework: the meaning must not be lost in each step. That's why we use RDF as a common language to describe every piece of the application. The W2000 model itself has been translated in RDF schemas with all the stereotypes fully described and ready to be directly referenced while building an application. We are persuaded that our RDF meta-description of W2000 will let us completely take advantage of the object oriented nature of the model itself and that this one can be the right choice, in this phase of W2000's research, in which we plan to get the methodology closer to a development-oriented point of view.

INTRODUCTION

Hypermedia and Web applications become more complex day after day, with new requirements and features. Every conceptual model, used to describe them, must change and evolve too. A very important consequence is that any change must be known by people using the model and working around it, and every tool or system based on it, made to support the design or the development, must be arranged to follow the changes with a controlled effort.

Our W2000 design methodology [1],[2] (developed at the HOC Lab of the Politecnico of Milan and at the SetLab of the University of Lecce) for the conceptual modeling of hypermedia and Web applications helps the designers to easily and fully express an idea and to discuss it with the customer, to manage the complexity and to completely describe the application and its behavior with different users and on different devices.

In this paper we talk about how the methodology, and the models it provides, can be represented with a layered approach like the one suggested by the W3C using RDF Schema [3] and RDF [4],[5], and how this representation can be useful going on with our research work around W2000.

This attempt wants to be a completion of what already done in [6] where a MOF [7] meta-model of W2000 was proposed, sharing most of the goals but pushing the results a little further. Our representation leads to machine readable schemas directly, letting us get much closer to the implementation and tries to fill the gap between design and development.

The W2000 design methodology

The explanation of the methodology in details is not a focus of this paper, but, in order to understand the topics we are going to talk about, it's necessary to know it in its fundamentals.

W2000 assumes that a good distinction between the different aspects of the application which have to be observed during the design is unavoidable, in order to make the design itself a structured and easily controllable process and to obtain a clear modeling, suitable for different users and delivery devices.

After the indispensable Requirement Analysis phase, carried ou according to a goal-oriented approach, the methodology suggests a sequence of steps briefly summarizable as follows:

- Information Design: the goal here is to describe the information that the application is going to deal with, giving it a structured organization. An important feature of this phase is that during the construction of the information structure the user's point of view [4] is held as fundamental.
- Navigation Design: this clarifies the most important aspect of hypermedia applications, reconsidering the information and its organization specifically from the viewpoint of its fruition and defining the navigational paths the user can follow.
- Publishing Design: the results of the previous steps must be completed with presentational considerations and organized into "pages" and "publishing units."
- Operations Design: this is the step in which all functional and transactional features (such as "register", "submit", etc.) not strictly connected to the hypermedia paradigm are modeled. Here the model allows the user to invoke the "functionalities" of the application.

Moreover, operations are the "building blocks" to be used to support complex transactions. The "customization methodology" comes together with the hypermedia design, to manage the different user-profiles and the various devices in many contexts.

W2000 has been applied to various contexts, widely accepted and used in several international projects, not only with research purposes.

The current state of our methodology, as described above, allows the designer to obtain a very advanced conceptual model of the applications before their development. The next phase of the research will try to make the abstractions concrete and to bring the methodology stage closer to those of development and implementation.

WHY A FORMAL DESCRIPTION

When a conceptual model evolves, all the tools created to support

the designers must be deeply modified but every needed change is often very difficult to be made, particularly when the tools belong to a complex complete design system based on a well structured design methodology.

A formal description like the one we propose gives a very sharp and detailed definition of the model, shared among the community, easy to be modified and with the precious property that every new feature can be quickly known and used by anybody.

The formal description we chose has some important features:

- it's sharp: the meaning is clear and well understandable for people already using the model or wanting to learn and use it;
- it's high-detailed: every aspect of the model can be covered by the description;
- it's flexible: any changes can be easily made to the model without problems;
- it's sharable: all the new releases of the conceptual model will be immediately available to all the designers' community;
- it's ready to be supported by tools: a meta-CASE [8] approach for the development of design tools can easily support changes and evolutions in the model with little effort;
- it's machine readable: not only design tools, but also development tools, for a direct prototyping of what designed, can be developed to read and interchange structured and well described information.

A MULTI-LEVEL APPROACH

The layered approach we present follows the one proposed by the W3C for the ontologies [9] and their descriptions, providing three abstraction levels from meta-schemes to schemes and then to data. In each layer the different concepts are firstly defined in structure and rules, then combined and applied to particular contexts and finally used and referenced to create the specific ontology.

Proper languages have been provided in order to express things at different levels of abstraction, to define concepts and constraints and to build schemes and complete descriptions. What links together the different layers is an instance mechanism which uses in a lower level what was defined in a higher one.

A structure like this one can be perfect for our purposes in the way that follows.

The W2000 methodology guides the designer in all the design phases and gives him/her suitable notations and graphical primitives (UML like) to express his ideas and conclusions. What results is a very detailed set of diagrams and descriptions, very powerful to be understood by developers and, obviously, by other designers, but absolutely not complete to be used by a prototyping tool. One of the directions of our research around W2000 is intended to create a complete framework which lets a W2000 designer describe an application in the way he is used to, but also gives as a result the direct prototype of the application itself, ready to be tested and used in most of its parts.

A description of the pure conceptual model can be organized at a Meta-Schema level, using an Object Oriented notation and representing it with a DAML+OIL [10] syntax and this can be used as an absolute reference of what the model provides (see Figure 1).

This representation will contain all the primitives of the model and all the rules and constraints they can be used above. Instances of such a structure can be exploited to build applications and to describe them as schemas in the Schema Layer. As a result every application will be a set of RDF Schemas, in a number enough to report what expressed in the Information Design, in the Navigation Design, in the Publishing design and so on... (see the W2000 methodology description above).

All the pure information associated with the application and all the actual pieces (with contents inside) composing the application itself will be described as instances of what defined above, in a suitable RDF format.

Figure 2 shows how what we explained can be applied to a real situation. The particular application displayed is part of an international project [11] and regards bank subjects: different credit cards had to be managed in a Web environment with several functionalities besides to the pure presentation. All the design phase has been done with the W2000 methodology but, for the actual implementation, standard and market technologies have been employed. The resulting product can absolutely be defined good but the effort to create it from phase to phase has been really considerable.

Of course it would be much better if most of the design effort could immediately result in something "concrete". This is what we want to get with the complete framework we are building around W2000, developing suitable tools which let the designer compose schemes and produce code. The multi-level structure we describe in this paper is the glue we chose to combine the pieces of the system in a whole: all the modules will communicate to one another referring to known concepts, definitions and constraints, expressed in a clear, coherent way.

As it can be easily argued, what represented in Figure 2 regards concepts and instances belonging to the W2000 Information Model, but similar structures can be built for the whole methodology. Obviously the different tools processing data from different models (Information, Navigation and so on) will make different use of them. For example, an Information scheme may result in a run-time database; a Publishing one will be used to build pages for a particular device.

Figure 1: The Multi Level Structure

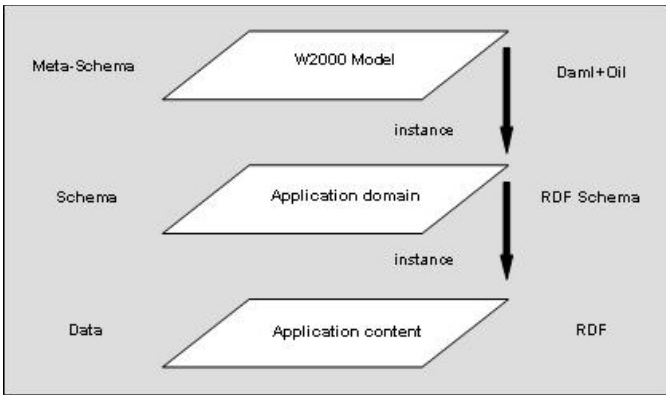
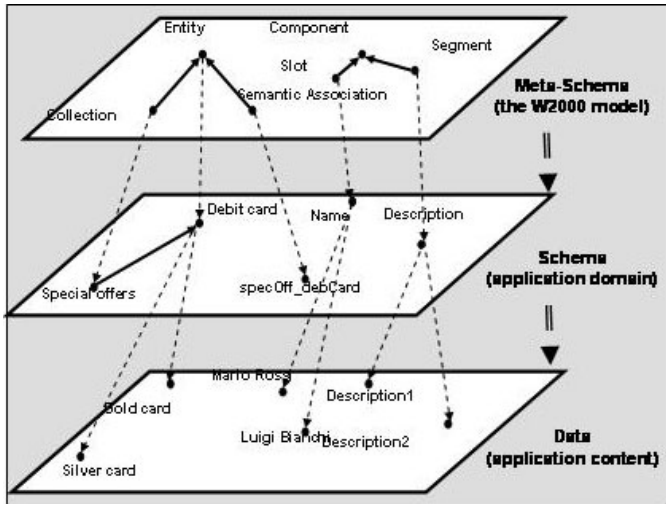


Figure 2: An Example



In the next chapter we expose the W2000 Information Model (see Introduction) in a deeper way, because we will describe the details of the "translation" process of a part of it. The same rules and conventions have been used to analyze the whole methodology.

Translating the MOF meta-model

In Figure 3 the MOF transcription of the W2000 Information Model is presented [6]. As already said, it is not part of what we propose in this study but it's one of our research group's results, granting precision, flexibility and methodological coherence, absolutely perfect to be "translated" in a DAML+OIL document.

It's not useful to explain all the scheme in details now but something may be remarked and used in the next reasoning.

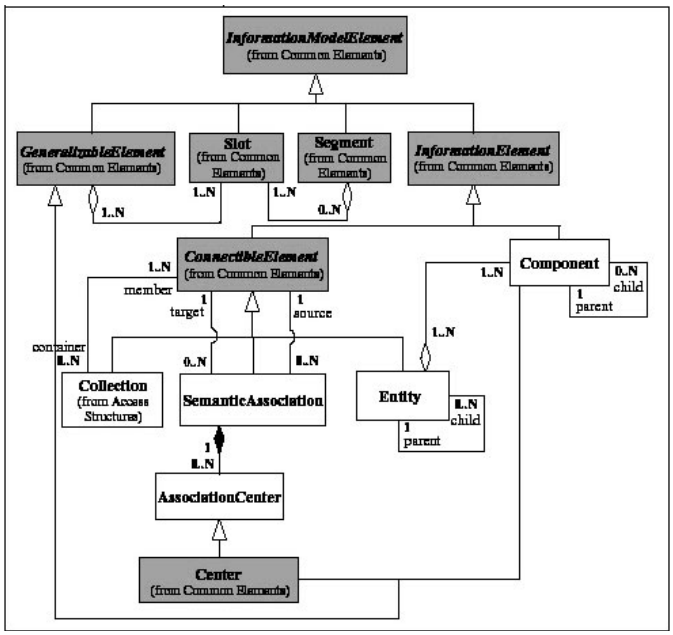
The key element is the *Entity*: it renders data of interest to the user as if they were conceptual objects (entities according to the ER jargon). An *Entity* resembles the concept of a class and, as classes, it can be the root of a generalization hierarchy (*parent* and *child* roles). An Entity is organized in semantic sub-units, called *Components*, which are pure organizational devices for grouping the contents of an entity into meaningful chunks. The result of this definition is a tree of components, based on the part-of relationship. Components can further be decomposed in subcomponents, but the actual contents can be associated with leaf components only. The contents of leaf components is defined in terms of *Slots*, i.e., the attributes that define the primitive information elements. A *Semantic Association* connects two *Entities* with a double meaning: it both creates the "infrastructure" for a possible navigation path (by connecting a source to a target) and has proper, local, information.

More general and abstract objects and their specializations are introduced every time it can simplify the scheme. Also a series of constraints is associated to schemes like the one presented.

All the structure above results in a DAML document which will be referenced as a namespace to build RDF-Schemas corresponding to complete models of applications.

An *InformationElement*, for example, is defined as a Daml Class and specified as a Subclass of the *InformationM_Element*. The same structure can be used to define Entities, Components and so on with all the needed constraints and *kind*, *cardinality* and *structure* specifications. Everything in the model can be properly translated in readable (by people and most of all by machines) code.

Figure 3: The MOF Representation of the W2000 Information Model



Classes and subclasses definitions let the hierarchical part of the resulting structure be reported in an immediate way, so that a *Component* can de defined in a few lines. Constraints and features can always be represented: different kinds of an *InformationElement* are defined suitably. Object properties help linking classes and adding constraints and extra features when needed.

The DAML document can be referenced as a namespace and all the concepts defined can be used in designing applications.

A meta-CASE design tool, programmed to understand and manage W2000 models, will let the designer build schemes and automatically produces the relative RFDS version.

The resulting document will be made of instances of the objects declared in the DAML model, with all the details and the values for any attributes and features.

For example, a *DebitCard* entity with a *Comment*, a *MinCardinality*, a *MaxCardinality*, an *ElementKind* (attributes) for the *Description* component can be easily declared. And in a similar way all the features of each element of the schema get a suitable form.

Once the application has been described in such a way and a RDFS schema is available, all the application objects may be built as instances in an RDF style to feed any prototyping device with ready-to-develop information.

CONCLUSION AND FUTURE WORK

The design phase for any kind of software is a very important but very expensive activity requiring good skills. Quality applications must be well thought and described before the development phase and a design methodology can be much more than a help in most occasions. It has been used to solve a large series of problems and collects a lot of experiences and knowledge.

That's why we think that powerful languages, mostly applied to knowledge matters, must be chosen also for more general applications.

An ontological approach, in facts, provides a standard, unique and reference representation for meta-data, ready to be shared and having an unambiguous meaning. And this is absolutely important, for example, when different people work on the same application during the same phase or in different moments.

In order to get the best results and to be sure that every design effort will give what the designer expects, we must be sure that the guiding methodology is fully understood by everyone. The meaning of each schema must be absolutely clear and this is what we primarily got with our work.

Our present work is trying to make the best use of what presented here and to exploit the advantages in our future work.

The output of each level is made of pure objects: O-O features like multiple inheritance can still be used to obtain different applications in the same family.

The RDF structure, providing *properties* referred to *resources* simplifies the application of several W2000 elements like the "customization rules", which support the multi-device and multi-user features of modern applications. Our future work will try to simplify the customization rules themselves and their application, also profiting by the new approach.

The description of the application domain which results can easily be used and managed in a complete *design-to-product* development framework.

Definitely, an RDF-based meta-description of W2000 will let us take advantage of the object oriented nature of the whole methodology, especially now that we are putting it closer to a development-oriented point of view.

REFERENCES

[1] L. Baresi, Garzotto, F., Paolini, P., and Valenti, S. (April 2000). **W2000: UML and HDM Provide a Unified Framework for Designing Web Application.**

[2] L. Baresi, Garzotto, F., and Paolini, P. (2000). From web sites

to web applications: New issues for conceptual modeling. **In Proceedings WWW Conceptual Modeling Conference,** Salt Lake City, October.

[3] Brickley, D. and Guha, R.V. Resource description framework (RDF) schema specification. Rec. W3C, March 1999. http://www.w3.org/TR/1999/PR-rdfschema-19990303.

[4] Lassila, O. and Swick, R.R. Resource description framework (RDF) Model and syntax specification. Rec. W3C, February 1999. http://www.w3.org/TR/1999/RECrdf-syntax-19990222.

[5] Brickley, D and Guha R.V, (February 1999). Resource Description Framework (RDF) Model and Syntax Specification, W3C Rec. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ (current May 2002).

[6] Baresi, L., Garzotto, F., and Maritati, M. (2002). W2000 as a MOF metamodel. **Proceedings of SCI 2002.**

[7] Boldt, J. Complete MOF 1.3 specification. http://www.omg.org/cgi-bin/doc?formal/00-04-03

[8] http://osiris.sunderland.ac.uk/rif/metacase/

[9] McGuinness, D. (August 14-18, 2000). Conceptual modeling for distributed ontology environments. **Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000).** Darmstadt, Germany.

[10] Connolly, D, van Harmelen, F., Horrocks, I, McGuinness, D., Patel-Schneider, P., and Stein, L. (March 2001) DAML+OIL. Reference Description, W3C Note 18 December 2001 (http://www.w3.org/TR/daml+oil-reference).

[11] http://www.uwaproject.org