Domain-Specific Instance Models in UML

Marco Torchiano[1] and Giorgio Bruno[2]

[1]Computer and Information Science Department (IDI) NTNU
Sem Sælands vei 7-9, N-7491 Trondheim, Norway,
Phone: +47 7359 4485 Fax: +47 7359 4466
Marco.Torchiano@idi.ntnu.no
[2]Dip.Automatica e Informatica, Politecnico di Torino
C.so Duca degli Abruzzi, 24, I-10129 Torino, Italy
Phone: +39 011 564 7003 Fax: +39 011 564 7099
bruno@polito.it

Abstract. UML is a widespread software modeling language. It can be a good candidate for modeling non-software systems, such as enterprise systems. While several proposals have been formulated, they are either incomplete or complex. In this paper we propose a modeling approach that is based on the basic object-orientation concepts. In particular we stress the use of instance models as a key concept.

1 Introduction

In the development of enterprise information systems, more time is spent in analyzing business needs than in coding [1]. Thus programming will increasingly deal with concepts at the application-domain level. Typical development will be relegated to small portions where it is unavoidable. It is important to find out if the Unified Modeling Language (UML) [4] (possibly extended) is suitable to satisfy modeling needs of users which are not software developers, e.g. process engineers.

Here we will present a solution for modeling complex enterprise systems that do not use the standard UML extension mechanism. Object-orientation offers a set of concepts, which could be effectively exploited without the need of extension mechanisms. In particular the proposed approach emphasizes the importance of instance-level models. We will take into consideration UML 1.3 since it is, at the time of this writing, the most widely know version. Much attention will be paid to supporting tools.

In section 2 we presents an assessment of UML capabilities, and in section 3 we analyze issues related to instance models. Then in section 3 we outline our proposal, and finally in section 5 we draw some conclusions.

2 Assessment of UML Capabilities

UML was not designed to model the enterprise domain, therefore it is important to to assess its capabilities in this area.

A. Banks Pidduck et al. (Eds.): CAISE 2002, LNCS 2348, pp. 774-777, 2002.

The organization of an enterprise can be modeled by means of the UML standard profile for business modeling. This profile defines a stereotype, named Organization Unit, which extends the Subsystem concept to represent the organization units of an enterprise. An example of organization model is shown in Fig. 1.

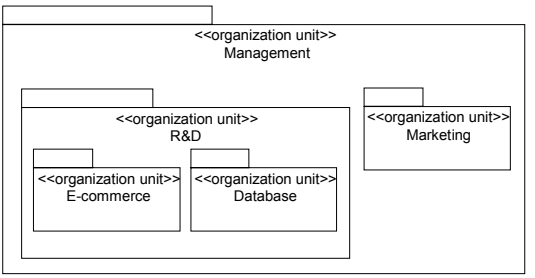

Fig. 1. Organization

In this example there is a top-level unit, *Management*, which contains two subunits, *R&D* and *Marketing*; the former contains units *E-commerce* and *Database*. This graphical representation in current tools does not have any semantics; it is a mere arrangement of shapes.

UML activity diagrams can be used to describe business processes. An example of a business process described with an activity diagrams is presented in Fig. 2.

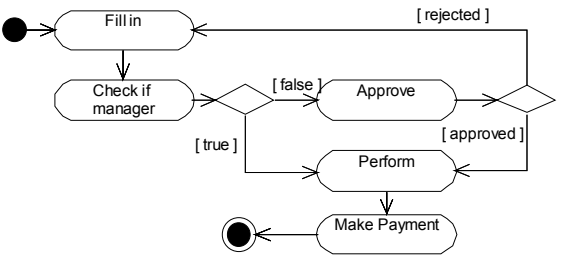

Fig. 2. Business Process

This is a simplified purchasing business process. After the purchase request has been filled in, if the requester is not a manager an approval is required, then the purchase can be performed and finally payment can be done. This diagram is not disciplined in any way and its business semantics is left largely unspecified.

A first and simple approach to enterprise modeling has been defined in the UML specification itself [4] by applying the UML Standard Profile for Business Modeling. It doesn't provide any support for business process modeling.

The UML profile for Enterprise Application Integration (EAI) [3] needs a part of business process modeling in order to set a base for integration. The models presented in reply to the initial OMG RFP are based on the existing UML modeling concepts and define a set of new modeling elements and relationships between them.

None of the previous works addresses in a satisfying way the description of business processes. In fact the OMG issued a request for proposal related to a UML Extensions for Workflow Process Definition [5].

According to [2] many of the useful business concepts are dispersed across different UML views and it is not easy to establish relationships between them. The following are areas of improvement: coordination semantics of business processes (e.g. explicit support for business events and exceptional situations), semantics of business roles and business rules.

3 Discussion of Instance Models

UML is class-centered. It is structured around class-relationship diagrams. Instance models are treated as second order entities; they are used only to show examples of data structures or to describe the behavior in particular scenarios

According to UML an object diagram is a graph of instances, including objects and data values. It shows a snapshot of the detailed state of a system at a point in time. The notation guide states that tools don't need to support a separate format for object diagrams. Class diagrams can contain objects, so a class diagram with objects and no classes is an "object diagram".

In many tools it is not possible to place an object in class diagrams, thus the only viable way to draw objects and links is to use a collaboration diagram, which is intended to model behavior and not static structure. Thus what we have is only a graphical diagram lacking any semantic foundation.

4 The Proposed Approach

Available UML tools at best provide only graphical support to model business concepts. They lack a sound link between the diagrams and a well-defined semantics.

We propose to use the basic concepts of object orientation to represent the modeling concepts required in different domains. Our proposal is intended to be an alternative to the approaches described in the section 2. The difference is mainly in the modeling mechanisms rather than in the semantics of the models.

The basic object-oriented concepts are classes and objects. Classes pertain to the level of problem domain, while objects pertain to the level of system domain. In the case of business models we can say that *process*, *activity* and *organization unit* are classes, while *MakePayment* is an object, i.e. an instance of a class (in particular of class Activity). Two sequential modeling activities are needed: domain modeling and system modeling.

The purpose of domain modeling is to analyze the problem domain in order to identify the main concepts (classes) and their relationships. The classes are abstract building blocks to be used in system modeling, while the relationships define the rules for how concrete building blocks (objects) can be connected to each other. In addition classes define attributes, i.e. they specify what kind of information can be stored into the concrete building blocks.

The aim of system modeling is to build a model of the intended system as a composition of building blocks, i.e. instances of the classes defined in the domain model. Connections between instances are regulated by the relationships between the corresponding classes.

Domain modeling and system modeling involve different roles. First, domain experts analyze the problem domain and, with the help a modeling expert, define a sort of modeling vocabulary (the domain model). Then, problem experts will use such a new "language" to build a model of the actual system.

A complex domain can present different features, which in most cases can be modeled separately. Therefore the domain model will result in a number of submodels that we call *schemas*. A schema is mapped into a UML package acting as a container of classes and relationships.

5 Conclusions

In the intention of UML creators, UML extension mechanism should be used in those cases when the basic expressive power of the language is not sufficient, and additional semantic information should be provided in order to provide modeling concepts at a higher abstraction level.

Several extensions in the field of enterprise model have been proposed or are the subject of ongoing efforts. None of them addresses the issue of a broad integrated enterprise model, which could provide the high level vocabulary to model enterprise systems.

We proposed an approach is easier to understand since it is based on the fundamental concepts of object orientation.

References

1. B. Bloom, J. Russell, J. Vlissides, M. Wegman. "High-Level Program Development", Dr. Dobb's Special Report December 2000
2. Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC). "UML Profile for EAI", document ad/00-08-07, August 2000.
3. Joint Submission "UML Profile for EAI", OMG document ad/00-08-05, August 2000
4. OMG. "Unified Modeling Language Specification" version 1.3, June 1999.
5. OMG, "UML Extensions for Workflow Process Definition - Request For Proposal" OMG Document: bom/2000-12-11