BUSINESS PROCESS MODELING WITH OBJECTS AND ROLES

Artur Caetano[*, †], António Rito Silva[*, ‡], José Tribolet[*, †]
[*]*Department of Information Systems and Computer Science, Instituto Superior Técnico, Technical University of Lisbon.*
[†]*CEO - Organizational Engineering Center, INOV, INESC Inovação.*
[‡]*Software Engineering Group, INESC-ID.*
Surface Mail Address: INESC, Rua Alves Redol 9, 1000-029 Lisboa, Portugal.

Keywords: business process modeling, role modeling, object-oriented analysis and design, organizational engineering.

Abstract: Role-based business process modeling deals with partitioning the universe of process modeling into different areas of concern by describing how business objects collaborate. A business object represents a concept of interest in the organization, such an activity or an entity, which can play multiple roles according to its behavior while interacting with other business objects. A specific business object collaboration can be expressed by the roles played by every participant in that scenario. This approach allows creating semantically richer business process models, and designing business objects where behavior is clearly separated and dependent on its usage context. Both of these results contribute to increase the understandability of process models and to enhance business object reuse.

1 INTRODUCTION

Organizational modeling deals with providing an enterprise-wide view of an organization from where decisions can be made. Business process modeling specializes on describing how activities interact with organizational entities in order to support the operation of the business. The analysis, modeling and representation of the knowledge about an organization and its processes has been the focus of specific research in past years and significant work has been done on developing business process modeling concepts, methodologies and ontologies as well as on the specification of process modeling languages.

As a product, business process models can be used for multiple purposes, such as facilitating human understanding and supporting process improvement, re-engineering and the analysis and design of process-oriented software implementations.

Business process models define a common medium for communicating organizational concepts, offering a set of domain level concepts and enabling a broader distribution of information among people with different knowledge about an organization. Business process analysis relies on a detailed description of process models and related concepts. In contrast, simulation allows for a detailed run-time breakdown of a process model, and does not rely on the structural properties of a business process but on the execution of previously designed processes using instances of entities and values. The combined results from both process analysis and simulation provide input for process reengineering, which involves redesigning the structural or collaboration aspects of a process model. Business process models may also be used to design business-driven software (Curtis 1992, Scheer 1999) or to derive workflow schemas (Aalst 2002).

Process modeling techniques often rely on capturing procedural and behavioral aspects of the business value chain, using data flow based models and are based on notations such as IDEF (McGowan 1993), where processes are described as a flow of activities along with the data and resources interchanged between these activities. However, and with such approaches, it is difficult to abstract away from the functional details of the process and to capture details such as how differently the same resource is used by different activities.

Modeling business processes involves capturing the structure of its business objects, their relationships and collaborations. A business object represents a concept of interest in the organization, such as human actors or automated actors, such as a production machine or an information system, and activities (whereas a set of activities coordinated towards the achievement of a goal is a business process). Identifying the business objects of an organization is fundamental to help documenting and evolv-

Caetano A., Rito Silva A. and Tribolet J. (2004).
BUSINESS PROCESS MODELING WITH OBJECTS AND ROLES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 109-114
DOI: 10.5220/0002648901090114
Copyright © SciTePress

ing the business by facilitating communication and analysis. Moreover, if an organization holds a documented view on its business objects and their relationships, then this information may assist later reusing the same business object across organizational units and in other business processes.

However, properly identifying the business objects of an organization is not a simple task, especially when reuse is a concern. For instance, the same business object may be manipulated by multiple different activities in different business contexts. On the one hand, each business object relates to a set of multiple other business objects. This leads to a highly connected relationship graph for every business object, which may not be easy to document or understand. On the other hand, a business object exhibits different behavior according to the relationships it has at a given time. For example, the same entity depicting a business product plays different roles when relating to a financial activity or to a manufacturing activity, meaning that the object's visible attributes, methods and behavior as a whole, may be different, depending on the object's active relationships. Furthermore, the specific usage context of a business object also defines it behavior. For example, in a given context, a business object may behave as an activity when being executed by an actor, and, in other context, it may behave as a business entity when being inspected by an audit activity.

Despite these issues, identifying an organization's business objects and specifying their behavior so that reuse is facilitated is fundamental to partition the universe of process modeling into different areas of concern, each of which can then be handled and documented independently. In order to address these problems, this paper proposes a set of organizational concepts, modeled as business objects, where their relationships are specified through the roles the objects play. A role describes the behavior of an object in a specific relationship and context, i.e., how an object is involved in a situation or what responsibilities it has. These concepts are represented by object-oriented constructs and are illustrated as a metamodel using the Unified Modeling Language (UML). This approach allows modeling a business process by (1) depicting the individual structure of business objects and (2) describing business object relationships according to the usage contexts of the objects.

The remaining of this paper is structured as follows: next section reviews role and business process modeling. Section 0 describes how role modeling can be used along with business process modeling to increase its expressive power along with some examples of application. Finally, section 0 outlines future work and draws some conclusions.

2 ROLE MODELING

From the perspective of sociological role theory, an organization is a system of interactions between entities constrained by norms and expectations. Entities can occupy a number of social positions and play the roles associated with these positions. Interactions are determined by the relationships among the roles, and constitute the structural aspect of the social system. They also include norms and rules designed to regulate the behavior of entities so that the goals of the system can be achieved. From this viewpoint, the analysis and design of an organizational system should focus on the three building blocks of a social system: the roles, the relationships among roles and the regulations that constrain them. Role theory defines concepts such as role and position in order to specify the organizational structure. In this perspective, Biddle and Thomas define a role as a collection of rights and duties relating to a position (Biddle 1979).

2.1 Roles and Software Engineering

Sociological role theory deals with collaboration and coordination of actors, focusing on the position and responsibilities of an element within an organization or system. Nonetheless, the concept of role is also a well-established modeling principle in computer science that aims at separating multiple crosscutting concerns existing in a given domain. It is used in methodologies such as RM-ODP (ISO 1995) and especially in object-oriented frameworks (Gottlob 1996, Kendall 1999, Halpin 2001). Here, a role is defined as a set of its properties which are important for an object to behave in a certain way as expected by other objects. Therefore, a role translates the expectations other objects have upon an object. Roles – just like objects – can be abstracted and later reused as types, since they capture the similar behavior properties of a class of individuals. Hence, there may be several instances of a role at a given time. Likewise, roles can also be decomposed and specialized and serve as a basis for reuse.

Roles emphasize on describing how objects interact with each other. While classes define the common capabilities of individual objects or instances, roles focus on the responsibilities of elements within a system or organization. A role model identifies the structure of elements and describes it as a structure of roles. A role model is similar to a UML collaboration diagram since both capture the interactions between objects in a given scenario. However, a UML collaboration diagram is subordinated to class diagrams and is based on instances of a specific application; its potential for reuse is thus

limited. Conversely, class diagrams address information modeling but not interaction modeling. Classes decompose objects based on their structural similarities and not because of their shared or collaborative activities and interactions. Role models overcome this limitation by describing a system in terms of their patterns of interaction, providing an abstraction that is orthogonal to classes and objects. However, they present a complimentary view on object interaction and do not aim replacing class models. Role models, like class models, can be instantiated, specialized and aggregated into composite models, promoting reuse in multiple contexts.

2.2 Roles and Process Modeling

In procedural and behavioral process models the activities that are to be carried out by an actor are spread around the process model because decomposition focus on function, i.e. activities are decomposed into a hierarchy of functionally simpler subactivities. Nevertheless, for an actor to carry out its activities, it needs to know what activities it must take part in, in what order those activities must take place, and what other actors or groups of actors it must interact with. Ould proposed Role-Activity Diagrams to overcome this issue (Ould 1995). Activities in a RAD describe the interaction between pairs of actor roles, from a driving to a target role. By executing an interaction activity, both of the interacting roles move to the next state in sequence. A RAD may also represent other activity flows than sequential, such as parallel and conditional. However, RADs only make use of a limited subset of the role concepts discusses before which somewhat limits the approach when used to capture context and describe object relationships.

First, actors are the only concept that may play a role. Thus, other concepts, such as entities, are not modeled according to the different roles they also play. Second, an actor role is defined by grouping the set of activities the actor can execute in some business process, thus describing its potential behavior. However, the same actor may execute different activities in different processes. By not capturing this, reuse is not promoted and only macroscopic roles are easily conveyed (e.g. an actor playing an accounting role may interact with actors playing the manufacturing and marketing roles; however the actor behaves differently in each case, playing different sub-roles of accounting). Third, roles, as used in RADs, are not abstracted as types or classes, which hold back role specialization and reuse.

Recent approaches, such as Eriksson's (2001), have explicitly integrated the object-oriented paradigm in business process modeling, making use of an extended subset of the UML as a modeling notation. UML activity and collaboration diagrams are used to represent the interaction between activities (objects and classes), grouped as roles (swim lanes). Activities, which are named with verbs, are descriptions of work that form one logical step within a business process. Activities are what organizational actors "do" in their roles. However, a role is used here with the same meaning of that of RADs, representing the macroscopic responsibilities of actors or of organizational parties, grouping the set of activities representing some unit of responsibility.

3 ROLE-BASED BUSINESS PROCESS MODELING

Role modeling has been adapted in software engineering as an abstraction mechanism to improve several analysis and design qualities of software when compared to standard object oriented techniques. These qualities include making models easier to understand and communicate, promoting reuse, and facilitating model adaptation. However, role model, as currently is used in business process modeling, is short of achieving the same goals since it is limited to grouping activities into roles, highlighting the responsibilities of each organizational actor or unit. Despite improving the understanding of how responsibilities are specified, it is not enough for capturing the context usage and behavior of business objects. For that reason, a challenge in process modeling is not only to understand how process activities are operationally carried out, but also to allow the universe of process modeling to the separated into different areas of concern, each of which can then be handled independently. This approach differs from current modeling approaches that focus on describing context free representations of processes. The remainder of this section describes the fundamental concepts of an object-oriented framework for role-based business process modeling.

3.1 Proposed Framework

An organization can be perceived as a set of business objects, coordinated towards the achievement of common goals. Business processes comprise a set of orchestrated activities, which operate over organizational entities, and are executed by human or mechanical actors in order to achieve goals.

Previously we have presented an object-oriented framework for capturing the structural and flow aspects behind business goals, activities and the supporting information system infrastructure (Vasconce-

los 2001, Caetano 2003). This paper presents a meta-model that allows describing the relationships of business objects in a specific interaction context. These concepts are represented as extensions to the UML as a set of stereotypes. Stereotype declarations are user-defined meta-elements that appear at the model layer of the UML four-layer meta-modeling hierarchy.

The core concept behind the model is that of business object. A *business object* represents a thing that is active in the business domain or organization. It is modeled as a class. *Business objects* may relate to other *business objects*. The semantics of a relationship is given by one or more *roles*.

A *role* defines the attributes or methods that are relevant to be stated for the *business object* to behave in a certain way as expected by other *business objects*, thus defining its observable behavioral aspect. *Roles* are organized into *role models.* A *role model* defines the set of *roles* required for a *business Object* to fulfill a collaboration.

An *activity* is a specialization of a *business object* and describes how to perform a piece of work. It corresponds to a verb in the business domain and is performed by at least one *actor*. A business process is an abstraction mechanism, comprising a set of *activities* that create a result with some value for an external or internal customer and contributes toward the achievement of *goals*.

An *entity* is a specialization of a *business object* and stands for a noun (e.g. product, document) or an *actor* in the business domain. Entities usually play the role of "resource" when relating to activities, representing their capacity to be created, accessed, modified, produced or consumed.

An *actor* is someone or something that can act in the context of an *activity*. It can be cognitive (a person) or mechanical (e.g. production machines and computer systems, including workflow systems).

A *goal* is a specialization of *entity* describing a measurable state that the organization intends to achieve.

A *constraint* asserts conditions over *business objects* or *roles*, defining their behavior and relationships (e.g. entity E may not play roles R and S simultaneously; activity A must be performed by actor M).

These concepts can be represented as extensions to the UML as a set of stereotypes, as shown in the following two diagrams. Stereotype declarations are user-defined meta-element that appears at the model layer of the UML four-layer meta-modeling hierarchy. A stereotype is declared by specifying its name, base class, tags and constraints (OMG 2003).

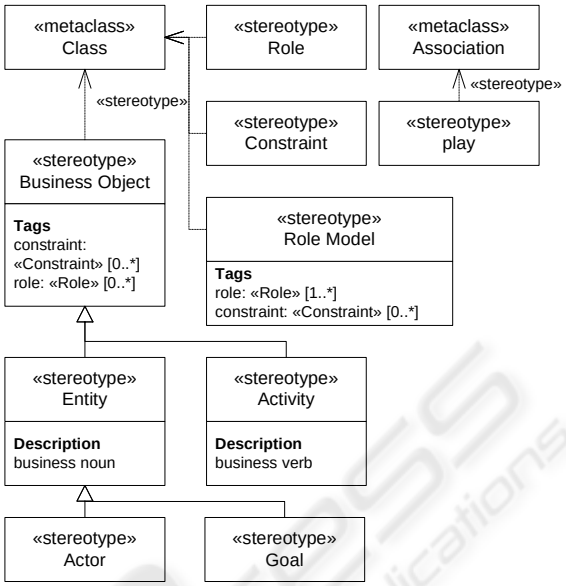

Figure 1. UML stereotype declaration.

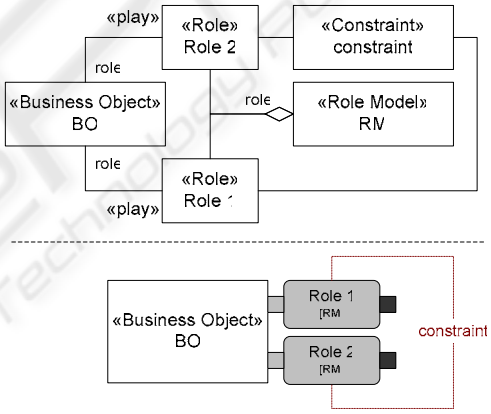

Figure 2. Roles, role models and business objects. Full notation (top). Compact notation (bottom).

Figure 1 represents the previously defined framework concepts as stereotypes. Note that not all tags and constraints are shown in the diagram for simplicity. Figure 2 shows the full and compact notation for associating business objects to roles on a specific role model.

3.2 Examples

This section exemplifies the framework concepts using two different scenarios. The first example show how the association entities and activities can be detailed. The second example focuses on a simplified business process, emphasizing entity and role modeling.

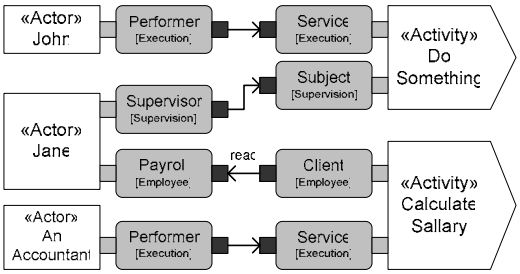

Figure 3. Actors and activities.

Figure 3 focuses on the relationships between three actors and two activities. Three role models are used by the business objects in this example: Execution, Supervision and Employee. A role model relates roles required for an object to express some behavior. For example, both actor John and actor Accountant are performing a Service role in an Execution role model.

Actor Jane is playing roles from two different roles models: supervision and employee. The Supervision role model relates the Supervisor and Subject roles (meaning that a business object that is able to play a Subject role can be supervised by some other business object). The Employee role model relates the Payroll and Client roles. Therefore, Jane is acting as an actor supervisor of the activity "do something". However, the same entity Jane is behaving as a resource when her payroll information is being accessed from the Calculate Salary activity. This means that in perspective of activity "do something", Jane is an actor, whereas from the "calculate salary" activity, Jane is being regarded not as an actor but as a payroll data resource.

Figure 4 depicts the top-level process for teaching a course without roles. The goal of the process is to enhance or add some skill to a person in a given subject. To do so, the instruct activity uses course material, is performed by an instructor and controlled by a course supervisor. As an input, the process takes an "unskilled" person and outputs the same person in a "skilled" state (these states as well as the corresponding transitions can be modeled with a state machine).

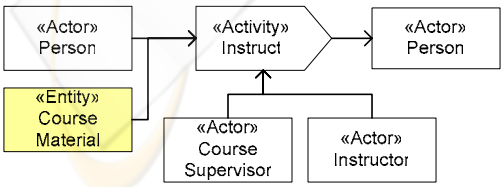

Figure 4. Top-level process for course instruction.

However, the above diagram can be enhanced with roles so that the behavior of each business object is made clearer, as shown in Figure 5.

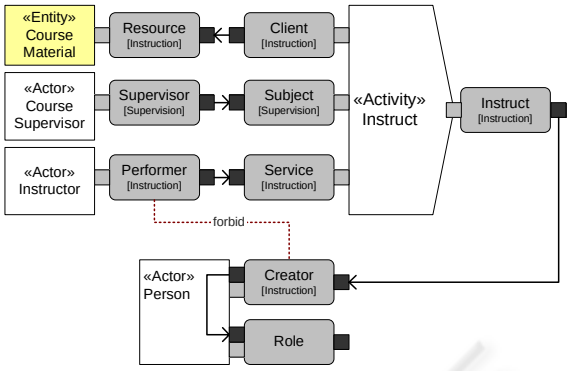

Figure 5. Top-level process for course instruction with the roles played by each business object.

The role diagram depicts that the instruction activity interacts with the trainee being instructed by creating a new role on that person. In case the instruction activity did not add a new role to the actor, then the situation would be modeled as the same described in Figure 3, where payroll information is read. In this case, the instruct activity would interact with the actor through a resource role that, in its turn would allow the actor's skills to be updated accordingly. A "forbid" constraint is defined between the Performer and Creator roles, meaning that the actor who is providing the services required for performing the instruction activity cannot be the same as the target of the instruction activity (i.e. the trainee cannot be the course instructor).

4 CONCLUSIONS AND FUTURE WORK

This paper has presented the fundamental concepts required for building a role-based business process model. These concepts were described as a UML meta-model using a set of stereotypes. To illustrate the concept usage, two examples were shown which focused separating different concerns while modeling business objects. The approach here presented relies on specifying individual business objects and making the collaborations between these dependent on the usage context. This is accomplished by defining and reusing roles that are assigned to business objects and composed in role models. The examples shown on this paper aim emphasizing that roles can be used to detail the collaboration patterns between business objects.

We are currently using these concepts in real organizations to enhance object-oriented representations of complex business processes. However, several areas that we are currently researching make direct usage of the concepts here presented. We emphasize activity-actor modeling and role reuse.

Activity-actor modeling deals with describing how an actor (such as a person, an information system or a web service) providing a set of services is supporting the execution of an activity, which, in turn, requires another set of services for successfully being completed. These service contracts can be specified as role models. A role binds an actor to an activity during a specific collaboration, while observing business constraints. A service is then a feature of an actor that enables her to execute an activity. The goal of such modeling is to provide the means to analyze the current situation of an organization and identifying requirements for future scenarios.

Role modeling can be used to promote reuse at role level instead of business object level. Since behavior is not activity-dependent, reuse is enabled at a role-level basis, as opposed to activity or process level as it often happens in process modeling. It is common that activities are identified and modeled by depicting either control flow (representing the orchestration of activities) or data flow. However, a resource is repeatedly used in multiple different contexts, in the same process or by different processes, which makes difficult identifying opportunities for reuse. This arises either from modeling the resource as a different business object on each different scenario it occurs, or because the resource ends up a specific and complex object. Moreover, resource or entity modeling is essential not only to business modeling but also during the requirements identification of the software that supports and evolves with the business. Entities or resources assigned to processes specify who has to work on the activity and what will be needed. However, current approaches to entity modeling offer only a set of simple features for the descriptions of resources and do not explicitly address separation of concerns to facilitate resource aspect reuse or minimizing and enclose the number and location of changes at the supporting information systems caused by business process redesign. For example, in UML, resources are not a specific language feature; and Scheer's (1999) event-driven process chains do not include construct for specifically modeling resources, which have to be modeled using entity-relationship diagrams. In this perspective, role modeling can facilitate resource and entity modeling and later reuse by identifying the set of intrinsic roles and role models.

REFERENCES

Aalst, W., Hee, K. (2002). *Workflow Management - Models, Methods, and Systems*. MIT Press.

Biddle B., Thomas E. (1979). *Role Theory: Concepts and Research*. Kluwer,.

Caetano A., Vasconcelos A., Sinogas P., Mendes R., Tribolet J., 2003. A Business Process Oriented Framework. *Information Resources Management Association International Conference*, Philadelphia, USA.

Curtis B., Kelner M., Over J. (1992). Process Modeling. *Communications of the ACM*, Vol. 35, No. 9, pp. 75-90, 1992.

Eriksson H., Penker M. (2001). *Business Modeling with UML: Business Patterns at Work*. OMG Press.

Gottlob G., Schrefl M., Röck B.m (1996). Extending Object-Oriented Systems with Roles. *ACM Transactions on Information Systems*, 14. 268-296.

Halpin, T. (2001). Augmenting UML with Fact-orientation. *Proceedings of the HICCS-34 Conference*.

ISO (1995). ISO/IEC 10746 International Standards Organization, ODP Reference Model, ISO.

Kendall E. (1998). Agent Roles and Role Models: New Abstractions for Multiagent System Analysis and Design. *International Workshop on Intelligent Agents in Information Management*.

Leymann F., D. Roller, M-T. Schmidt (2002). Web Services and Business Process Management. *IBM Systems Journal*, Vol. 41, No. 2.

McGowan C., Bohmer L. (1993). Model-based business process improvement. *15th International Conference on Software Engineering*, IEEE Computer Society Press.

OMG (2003, March). *Unified Modeling Language Specification*. Version 1.5, formal/03-03-01.

Ould M. (1995). *Business Processes: Modeling and Analysis for Reengineering and Improvement*. John Wiley and Sons.

Scheer, A.-W. (1999). *ARIS - Business Process Modeling*. 2nd edition, Springer.

Vasconcelos A., Caetano A., Neves J., Sinogas P., Mendes R., Tribolet J. (2001). A Framework for Modeling Strategy, Business Processes and Information Systems. *EDOC 2001, IEEE* Computer Society Press.