

A Role-Based Framework for Business Process Modeling

Artur Caetano^{1,3}, Marielba Zacarias¹, António Rito Silva^{2,3}, José Tribolet^{1,3}

¹ Organizational Engineering Center, INESC INOV.

² Software Engineering Group, INESC ID.

³ Department of Information Systems and Computer Science,
Instituto Superior Técnico, Technical University of Lisbon.

E-mail: artur.caetano@inesc.pt, mzacaria@ualg.pt, antonio.silva@inesc.pt, jose.tribolet@inesc.pt

Abstract

Business objects are object-oriented representations of the concepts of interest in an organization, such as activities, resources and actors. Business objects collaborate with one another in order to achieve business goals, showing different behavior and properties according to each specific collaboration context. This means the same business object may be perceived differently depending on the business objects it is collaborating with. However, most approaches to business process modeling do not separate the collaborative aspects of a business object from its internal aspects. To cope with such issues, this paper makes use of role modeling to separate these concerns while increasing the understandability and reusability of business process models. This approach makes use of object-oriented concepts to separate a business process model into a business object model and a role model. The business object models deals with specifying the structure and intrinsic behavior of business objects, while the role model specifies its collaborative aspects.

1. Introduction

Business process modeling specializes on describing how activities interact and relate with other organizational elements while supporting the operation of the business. The representation of an organization and its processes has been the focus of 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 [10, 18, 19, 20]. Business process modeling can also be used for multiple purposes, such as facilitating human understanding and communication [31], supporting process improvement and re-engineering through business process analysis and simulation [9, 19], automating the execution of business processes [1,25] and facilitating coordinated business and system development by keeping the alignment between processes and their support systems [6].

Modeling business processes requires capturing the essential concepts that are involved in a collaboration, their responsibilities and the business meaningful relationships between them. Examples of these concepts are activities,

the resources used by activities and the human or automated actors who perform these activities. Identifying the properties and relationships of these concepts is fundamental to understand and evolve the business by facilitating model and concept communication between stakeholders, business and system specialists. It also promotes concept reuse across business processes. We model the business concepts, their responsibilities and relationships as classes of business objects. This helps building up a consistent object-oriented glossary of concepts that can later be composed and specialized.

Characterizing the type, properties and relationships of a business object is not straightforward since the business concepts the objects are modeling are used in different business contexts, meaning that the same business object may relate to several other business objects in the organization. For example, the business object modeling a Product may be brought into play in several processes, such as Manufacturing, Logistics and Selling, and in each of these processes it will relate with different activities and resources. Analyzing this relationship network may not be easy because the object acts as a multi-dimensional entity depending on the context of usage. A Product property such as Location, while fundamental to the Logistics process, may not be relevant to the Manufacturing process or may even have a different meaning in that context. The same reasoning applies not only to the properties of the object but also to its behavior.

Some coordination-oriented process modeling techniques, such as Role Activity Diagrams [24], provide the means to identify roles and interactions. Roles organize a process' activities into sets of operations associated with a given participant in the process. Interactions show the dependencies between those participants. While this approach improves the understandability of a process model since it depicts what a participant does in a process, it falls short explaining the behavior of the business objects of the process. These techniques do not make explicit what behavior a specific object is displaying in a specific context of interaction and why is it displaying such a behavior. This would facilitate reusing the object in a different context. Additionally, roles are considered collections of activities and not types, so they can not be composed and specialized.

Properly identifying the behavior and properties pertaining to a collaboration is important to describe a business object in an organizational setting so that it can be understood and later reused. The importance of this becomes even greater when it is a goal to keep the fit or alignment between the process model, the environment and the supporting systems level. If business objects are modeled at business process level as flat entities, i.e. without its properties and behavior being described as dependent on the context, then the objects will have no explicit information on how to guide the design of a business support system that is able to cope with evolution and change. For example, if the Manufacturing process changes, there may be changes to the Product object. However, if the supporting systems have represented the Product object without separating its different concerns, then there will be no explicit information on the properties and behavior that will require modifications.

This paper focus on describing on how a method to separate business process modeling and its business objects into different areas of concern, each of which can be handled independently and later composed to synthesize a complete model. To do so, we propose defining two complementary conceptual models, a and a business object model. The role model describes business object collaborations and the properties of business objects that are concerned with each role, being each role a type on its own. The business object model describes the structure and the properties of a business object that context-free. The relationships between business objects are specified by the roles that the objects play in a collaboration. We argue that using roles and business objects to model business processes improves the understandability of the individual business objects and of the process model as a whole. It also improves model reengineering by facilitating model reuse and making explicit the dependencies between the model elements.

The remainder of this paper is structured as follows: next section reviews some of the research on business process modeling. Section 3 reviews role modeling, describes how roles can be identified and defines the concepts of business objects and role. Section 4 presents how the business object and the role model can describe a business process, followed by an example of application in section 5. Finally, section 6 sets out the paper conclusions and outlines future work.

2. Business Process Modeling

The Workflow Reference Model [34] defines a business process as a set of one or more linked procedures or activities that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships. This definition extends the definition proposed by Davenport and Short [8], stating that a business process is a set

of logically related tasks performed to achieve a defined business outcome. Most approaches to business process modeling focus on some sort of process diagram, which show how activities are coordinated in the course of a business process. Indeed, there is little disagreement about the key elements of a process diagram: it must contain activities and activity connectors. In addition, there are usually ways to represent decision points, and ways to express various activity coordination patterns, such as sequential flow, branching and parallel execution. Some techniques introduce swim-lanes to indicate the responsibilities of participants, such as departments or individuals, thus representing the activities these actors perform in the context of the process. Two representative coordination-oriented techniques that make use of these concepts are Role Interaction Networks [27] and Role Activity Diagrams [24]. Activities describe the interaction between pairs of actor roles, from a driving to a target role.

The Object Management Group has recently summarized the basics of process diagrams in its upcoming UML 2.0 activity diagram notation [22]. Similarly, the BPMI working group has just completed BPMN, a notation that is designed for describing processes in business process diagrams [5]. However, at a high-level of abstraction UML 2.0 activity diagrams and BPMN are quite similar. In addition to these notations, there are others like the IDEF family and the proprietary notations supported by process and workflow products.

However, business process modeling is not limited to process diagrams. The focus of this paper is not on process diagrams but on the description of the roles that are used to specify the responsibilities of business objects. A business object is the model of a concept in the business universe of discourse. It plays roles in a business process by means of participating in different activities. Business objects participate in different business processes in different contexts, thus playing multiple roles. It is important to note that process diagrams do not fully describe the business object structure and relationships, and do not emphasize how goals are achieved by performing an activity. Besides, they only identify actor roles, i.e. the roles of the performer of an activity. This means, for example, that the properties of a resource that is used by multiple activities are not separated according to its usage context.

3. Role Modeling

Role theory started to generated interest among social scientists from many backgrounds, such as psychology and sociology, in the late 1920s. Its central concern has been with patterns of human conduct and with context and social structure as well as with individual response. The motivation for roles is to allow particular viewpoints regarding the factors presumed to be influential in governing behavior and lies on a theatrical analogy of actors playing parts or roles in a play, as Biddle and Thomas [7]

have stated: “When actors portray a character in a play, their performance is determined by the script, the director’s instructions, the performances of fellow actors, and reactions of the audience as well as by the acting talents of the players. Apart from differences between actors in the interpretation of their parts, the performance of each actor is programmed by all of these external factors; consequently, there are significant similarities in the performances of actors taking the same part, no matter who the actors are.”

There exist multiple definitions for the concept of role. For example, in the late 1970s, sociological role theorists defined a role as “a comprehensive pattern for behavior and attitude” [29] or as “behavioral repertoire characteristic of a person or a position” [3]. However, there is still no consensus on the definition or properties of role. Nonetheless, the concept of role is used in computer science and software engineering as a modeling technique that supports the separation of concerns, i.e. the separation of the “behavioral repertoire characteristics” of some concept. It is being used in methodologies such as RM-ODP [14] and in several object-oriented frameworks [10, 13, 26]. Kristiansen [16] has proposed a set role properties, which are commonly regarded as a conceptual basis for defining roles in software engineering.

In business process modeling there are also approaches based on role modeling. Despite being quite similar, we emphasize Role Interaction Networks [27] and Role Activity Diagrams [24]. Here, roles are considered as sets of ordered interactions. Activities describe the interaction between pairs of actor roles, from a driving to a target role. However, these approaches do not fully depict context, describe object relationships or separate other concerns than actor roles.

3.1 Business Objects and Roles

Modeling consists of identifying things of interest in a given universe of discourse and representing them in a model. In business modeling, the universe of discourse corresponds to what is perceived of an organization as being reality by business domain experts. Ontologies typically distinguish entities from activities in the universe of discourse.

Entities are things that exist in the business, either concrete or abstract. Activities are things that happen in the business making use of entities. Both of these concepts are modeled as specialized business objects. A business object is then the super type of all objects that represent business concepts with a well-defined boundary and identity, and encapsulate properties such as its definition, attributes, behavior and relationships [23, 24]. The state of a business object is characterized by its attributes and values. The behavior are the actions a business object is capable of performing to fulfill its purpose, including chang-

ing its intrinsic attributes and collaborating with other business objects.

Business objects have intrinsic features that describe it in isolation and extrinsic features that arise from the relationships with other business objects. For example, a Person has intrinsic features such as Age and Sex, and extrinsic features such as job position and Salary which derive from the relationship between the Person and an Organization. Intrinsic features may change over time (e.g. Age) but always characterize the object. However, extrinsic features may eventually become inappropriate or conflicting (e.g. Salary of an unemployed person, Salary of a Person with multiple jobs).

One way to separate the intrinsic features from the extrinsic features of an object is by the usage of roles [4, 16, 26]. Roles, as a modeling construct, aim at separating the concerns that arise from business object collaborations. We define a role as the observable behavioral of a business object defined in a specific collaboration context. Thus, a role represents the extrinsic features of a business object when it participates in some activity.

3.2 Identifying Roles

To distinguish roles from natural types or entities, Guarino *et al.* proposed two criteria [12]. These criteria assert that a role is founded and lacks semantic rigidity. Something is founded if it is defined in terms of relationships with other things in a given context. For instance, the concept of Reader is founded since for a Person to be a Reader, there must be something being read. Conversely, a Person is not founded, since its intrinsic properties (e.g. Age, Sex) are defined in their own regardless of the collaborations with other things. Something is semantically rigid if its identity directly depends on being kind of some class. A Book is semantically rigid since its identity is still that of a Book regardless someone is reading it or not. In contrast, Reader is not semantically rigid, since an entity filling the role of Reader retains its identity outside the context of that role. For example, a Person is a Reader while reading a Book, but when it stops reading it, it is still a Person. Therefore, roles are founded and semantically non-rigid types while entities are non-founded and semantically rigid types.

4. Role-Based Business Process Modeling

Our approach to business process modeling deals with decomposing the modeling universe into two complementary models, the business object model and the role model, and binding these two models into an integrated specification of the business. The business object model deals with specifying the structure and intrinsic properties of business objects. At this level of abstraction, an organization is modeled as a network of business objects. However, business objects relate to other business objects in

specific contexts and are often used in more than one context, where they may play different roles. So, the roles for a business object only need to be included in its definition when the object acts in the collaboration contexts described by the roles. It is also impossible to forecast the possible roles for a business object, and adding superfluous roles impairs several design quality attributes such as understandability, maintainability and reusability. To deal with such issues, roles and business objects should be dealt with separately, and later bound together.

The concept of role allows a system to be decomposed into a set of business objects capable of clearly separating core parts and collaboration-dependent parts, and then to abstract and compose such objects. Consequently, a set of roles helps business objects to be defined to be more reusable and extensible, and the roles become more reusable as a unit encapsulating specific collaborations. Roles are organized into role models, which deal with specifying the network of related roles required for a collaboration to happen. The global business process model then consists of these two complementary models, the business object model and the role model.

We define and represent both of these models using the Unified Modeling Language [21] since its graphical syntax and semantics is well-known. However, the standard UML does not have explicit constructs to represent the required business domain concepts. Thus, we make use of the UML extensibility package to define such concepts. The extensibility package specifies how UML model elements can be extended and customized with new graphical representations and new semantics by specifying stereotypes, tagged values and constraints. A coherent set of such extensions defined for a specific purpose makes up a UML profile [2, 21]. The next subsections describe how the business object models and role models are represented.

4.1 The Business Object Model

The business object model specifies the structure and intrinsic properties of business objects. Business objects are coordinated towards the achievement of goals, being the coordination mechanism the business process.

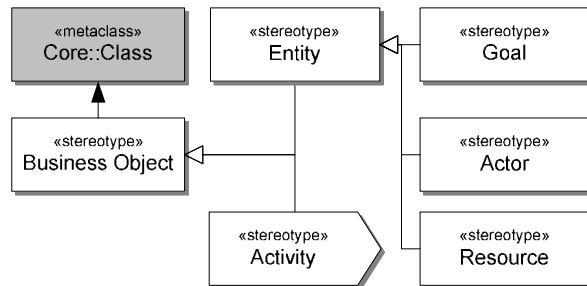


Figure 1. Classes in the business object profile.

A business process is composed of *activities* that use *resources*, such as materials or information, to produce

output *resources*, thus achieving *business goals*. The activities of a business process are performed by one or more *actors* or business support systems, which represent people, systems or a combination of both. At a large scale, business processes are composed as value chains that produce value to external customers [28]. Figure 1 is a class diagram describing the UML stereotypes that are used in the business object model. Table 1 further describes these stereotypes.

Table 1. Business object stereotypes.

Stereotype	Base	Description
<i>Business Object</i>	–	A concept of interest in the organization.
<i>Activity</i>	<i>Business Object</i>	Business verb. Describe how a piece of work is performed. Are performed by <i>Actors</i> , and operate over <i>Resources</i> .
<i>Entity</i>	<i>Business Object</i>	Business noun. Describe concrete or abstract business concepts of interest and participate in multiple business collaborations.
<i>Resource</i>	<i>Entity</i>	Input or output of an <i>Activity</i> .
<i>Actor</i>	<i>Entity</i>	Someone (human actor) or something (automated actor, such as an information system or a production machine) that can perform the actions required by an <i>Activity</i> .
<i>Goal</i>	<i>Entity</i>	A measurable state that the organization intends to achieve. <i>Goals</i> are achieved by <i>Business Objects</i> , especially <i>Activities</i> .

A business object is a UML class. Therefore, business objects, such as activities and resources, may be aggregated, composed and generalized. Business object models are represented as UML class diagrams and the intrinsic behavior of the objects is represented using UML's behavioral diagrams. However, no collaborations between business objects are represented in this model but in the role model.

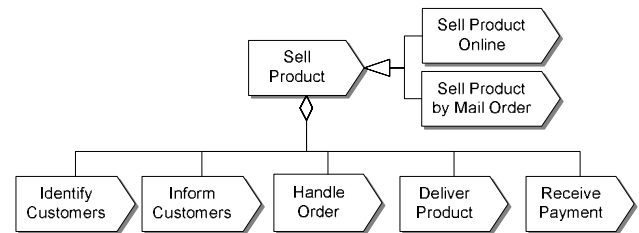


Figure 2. Example of activity decomposition and specialization.

Figure 2 shows an example of a class diagram depicting composition (vertical relationships) and specialization (horizontal relationships). Each icon represents an activity as defined above. The Sell Product activity is composed by a set of sub-activities such as Identify Customers and Handle Order. These activities can be further decomposed into smaller actions. Sell Product is specialized into two activities, Sell by Mail Order and Sell Online. Specialization confines the domain of usage of an activity.

4.2 The Role Model

Roles are a separation of concerns mechanism that allows business objects to be observed from different perspectives. For example, the attributes of a Book business object may be perceived differently by different activities (v. Figure 3). While the Title property maintains its meaning, Cost represents the production cost to the Manufacturing activity while it represents the book price to the Marketing activity. Attributes such as Style do not make sense in the context of the Manufacturing activity.

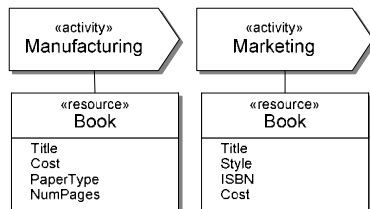


Figure 3. The Book business object as seen by two different activities.

Role models identify roles as types and describe the network of roles required for a specific collaboration to happen. As a player of a collaboration, a role defines the set of extrinsic properties and behavior necessary to realize its participating collaborations.



Figure 4. Representation of the role model package (left) and roles (right).

Role models are represented as UML packages with two compartments (v. Figure 4, left). The bottom compartment of the role model (role behavior) is a standard UML activity or interaction diagram describing how the roles are orchestrated. The top compartment of the package (role collaborations) depicts how roles are related within the role model. Roles are represented by ovals and the arc is a navigable collaboration relationship between the roles. A role representation includes its name on the top of the oval. Optionally it may also indicate in parenthesis the name of the role model to where the role belongs so that its scope is clearly defined (v. Figure 4, right).

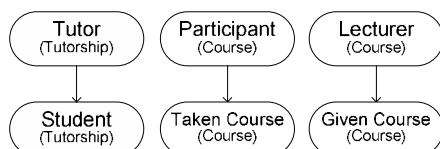


Figure 5. Example of role collaborations.

Figure 5 shows an example of three role collaborations contained in two role models. The Tutorship role model defines a collaboration pattern between two roles Tutor and Student, whereas the Course role model defines two

pairs of collaborations: Participant/Taken Course and Lecturer/Given Course.

Roles are modeled as classes and represented in class diagrams. Methods and attributes concerning the specific collaboration context can be included in this class diagram. Roles can also be constrained. A constraint asserts conditions between the roles in a role model and can be expressed informally or formally. An example is disallowing two roles to be played simultaneously by the same player, such as forbidding an object playing the role of Tutor and that of Student in the same collaboration context.

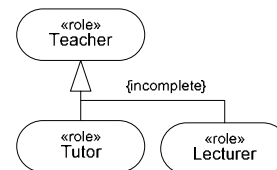


Figure 6. Example of role specialization.

Figure 6 shows a class diagram depicting the specialization of the Teacher role as Tutor and Lecturer. As expected, specialization here means that if a business object is able to play a child role it is also able to play the behavior of the parent role.

4.3 Binding Roles to Objects

Roles are bound to business objects pertaining to a specific business object model. The binding is accomplished via the «play» relationship stereotype that links a business object to a role and it means the source business object is able to exhibit the behavior stated in the target role.

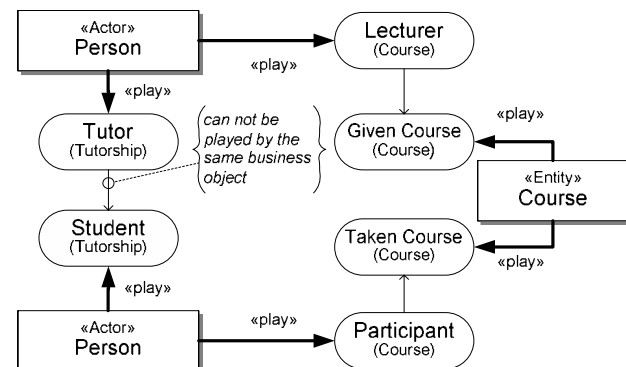


Figure 7. Binding roles to business objects.

Figure 7 shows a class diagram where the roles Tutor/Student, Lecturer/Given Course and Participant/Taken Course specified earlier are bound to two types of business objects, Person and Course. The business objects are depicted as shaded rectangles and the binding as a strong arrow. Light arrows represent role collaborations. A constraint defined in the Tutorship role model is also shown here. It asserts that the instances actually playing the Tutor and Student role must be different objects.

5. Example

This section extends the example introduced in the previous section. Figure 8 depicts a simplified business process for instructing a course. The goal of this process is adding a skill to a Participant. A skill is an organizational resource that can be articulated as role through an actor in the context of an activity. The Instruct activity requires course information, represented as the Course entity, and it is performed by a Lecturer. The process is controlled by a Course Supervisor. The main input of the activity is a Participant actor. If the activity succeeds, it changes the state of the actor so that it is able to play a new role.

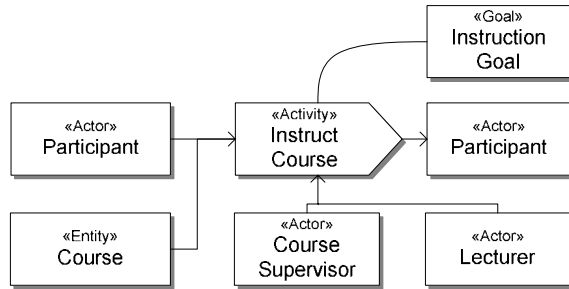


Figure 8. Top-level business object model for course instruction.

Figure 9 and Figure 10 illustrate the model where a set of roles has been bound to the business objects, defining the collaborations between actors and resources in the context of the Course and Supervision.

The interaction between the Instruct activity and the Participant actor results in a new skill, modeled as a role. This is represented as the result of the collaboration between the Participant actor and the Course that enables the Instruct activity to add the new role. Two constraints are also defined, asserting that the Participant and Lecturer roles cannot be played by the same business object and that the course Supervisor may not be the course Lecturer.

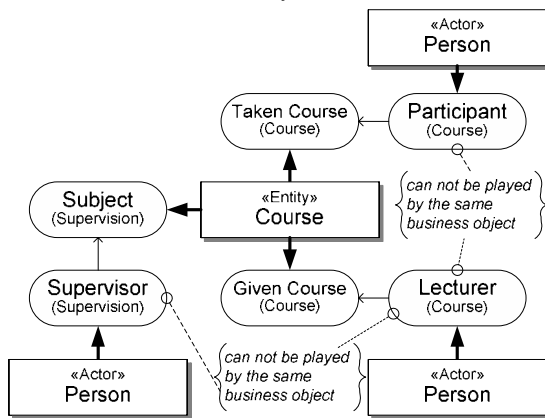


Figure 9. Role binding.

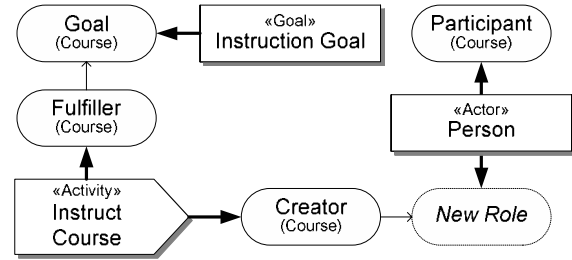


Figure 10. Adding a new role to an object.

6. Conclusions and Future Work

This paper has introduced a set of fundamental concepts towards a conceptual object-oriented framework for role-based business process modeling. It relies on defining two distinct models: the business object model and the role model. The first focus on the description of business objects, i.e. the components of a business. It represents the type of each business object, its intrinsic behavior and properties but does not address the representation of the object's collaboration-related features. The role model specifies roles as types that can be specialized and aggregated. Role reuse is possible whenever the semantics of the interaction pattern is the same. The role model depicts the collaborative behavior between roles and the constraints that regulate them. Roles are bound to business objects in a specific business object model that defines their usage context.

This approach separates the specification of the intrinsic features of a business object from its extrinsic features, i.e. the properties and behavior that arise from the collaborations with other objects. This increases the understandability of the business process since each different aspect of the business object may be discussed, analyzed and dealt with separately. Additionally roles also contribute to keep the alignment between the multiple organizational levels where the business process is defined. When a business object specified at business level is mapped to a component at business process support systems level, roles provide information on the component so that changes to any of the organizational levels can be traced. Since the collaborative aspects of a business object are specified outside the object as roles, changes to a business process only interfere with the roles which derive from the corresponding activities, leaving the intrinsic properties of the object and its remaining roles unchanged. This means that only the implementation of the concerned roles needs modifications. The same reasoning applies the other way around, when the implementation of a specific role or business object requires technical modifications due to the evolution of the software.

We are currently extending this framework to use role modeling to enhance the representation of the interaction between business objects and the corresponding business support systems. In the case of an activity, this means

specifying the required roles for it to be performed by an actor. At business support systems level, it means making explicit the roles that the organization can actually provide in order to perform or support the activity. Our aim is to analyze the gap between the existing human skills and information system services of an organization and the requirements imposed by the as-is and to-be business models so that the alignment between these two levels may be improved.

References

1. W.Aalst, K.Hee, *Workflow Management*, MIT Press, 2002.
2. S.Alhir, *Unified Modeling Language Extension Mechanisms, Distributed Computing*, 1998.
3. C. Bachman, M. Daya, *The role concept in data models*, Proceedings of the 3rd International Conference on VLDB, 1977.
4. B. Biddle, E. Thomas, *Role Theory, Concepts and Research*, Kluwer Publishers, 1979.
5. BPMI, *Business Process Modeling Notation*, Version 1.0, 2004 (from www.bpmn.org)
6. Y. Chan, *Why Haven't We Mastered Alignment?: The Importance of the Informal Organization Structure*, MISQ Executive, Vol.1, No.2, 2002.
7. B. Curtis, M. Kelner, J. Over, *Process Modeling, Communications of the ACM*, Vol. 35, No. 9, 1992.
8. T. Davenport, J. Short, *The New Industrial Engineering: Information Technology and Business Process Redesign*. Sloan Management Review, 1990.
9. H. Eertink, W. Janssen, P. Lutthuis, W. Teeuw, C. Vissers, *A Business Process Design Language*, World Congress on Formal Methods, Springer, 1999, pp. 76-95.
10. H. Eriksson, M. Penker, *Business Modeling with UML*, OMG Press, 2001.
11. G. Gottlob, M. Schrefl, B. Röck, *Extending Object-Oriented Systems with Roles*, ACM Transactions on Information Systems, Vol. 14, 1996 pp. 268-296.
12. N. Guarino, M. Carrara, and P. Giaretta. *An ontology of meta-level categories*. Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning, pages 270–280. Morgan Kaufmann, 1994.
13. T. Halpin, *Augmenting UML with Fact-orientation*, 34th Hawaii International Conference on System Sciences, IEEE Press, Hawaii, USA, 2001.
14. ISO, *ISO/IEC 10746 ODP Reference Model*, International Standards Organization, 1995.
15. E. Kendall, *Agent Roles and Role Models, New Abstractions for Multiagent System Analysis and Design*, International Workshop on Intelligent Agents, 1998.
16. B. Kristiansen, *Object-Oriented Modeling with Roles*, 1st International Conference on Object Information Systems, 1996.
17. F. Leymann, D. Roller, M. Schmidt, *Web Services and Business Process Management*, IBM Systems Journal, Vol. 41, No. 2, 2002.
18. M. Madhavji, *The Process Cycle*, Software Engineering Journal, Vol. 6, No. 5, 1991.
19. C. McGowan, L. Bohmer, *Model-based business process improvement*, 15th International Conference on Software Engineering, IEEE Computer Society Press, 1993.
20. D. Miers, *Business Process Engineering*, C-T Colin, Kogan Page, London, 1996.
21. OMG, *Unified Modeling Language Specification*, Version 1.5, formal/03-03-01, 2003.
22. OMG, *Unified Modeling Language: Superstructure*, Version 2.0, Final Adopted Specification, ptc/03-08-02, 2003.
23. OMG, *Business Object Management Special Interest Group (BOMSIG) Glossary of Terms*, 1995.
24. M. Ould, *Business Processes, Modeling and Analysis for Reengineering and Improvement*, John Wiley & Sons, 1995.
25. A. Scheer, *ARIS – Business Process Modeling*, 2nd edition, Springer, 1999.
26. T. Reenskaug et al., *Working With Objects: The OOram Software Engineering Method*. ManningPublication Co., 1996.
27. B. Singh, G. Rein. *Role Interaction Nets (RINs): A Process Description Formalism*, MCC, 1992
28. D. Taylor, *Business Engineering with Object Technology*, John Wiley & Sons, 1995.
29. R. Turner, *Strategy for Developing an Integrated Role Theory*. Humboldt Journal of Sociology and Religion 7: 123-139, 1979.
30. M. Uschold, M. King, S. Moralee, Y. Zorgios, *The Enterprise Ontology*, The Knowledge Engineering Review, Vol. 13, 1998.
31. E. Verharen, *A Language-Action Perspective on the Design of Cooperative Information Agents*, CIP-Gegevens Koninklijke Bibliotheek, 1997.
32. T. Walford, *Business Process Implementation for IT Professionals and Managers*, Artech House, Norwood, MA, 1999.
33. E. Yourdon, *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
34. Workflow Management Coalition, *The Workflow Reference Model*, 1995.