USING ROLES TO SPECIFY BUSINESS OBJECT COLLABORATIONS

Artur Caetano *†

António Rito Silva *‡

José Tribolet *†

*Department of Information Systems and Computer Science, Instituto Superior Técnico, Technical University of Lisbon.
† Center for Organizational Engineering (CEO), INOV, INESC Inovação.
‡ Software Engineering Group, INESC-ID.
Surface Mail Address: INESC, Rua Alves Redol 9, 1000-029 Lisboa, Portugal.
E-Mail: { artur.caetano, rito.silva, jose.tribolet }@dei.ist.utl.pt

ABSTRACT

Role-based business process modeling deals with separating the universe of process modeling into different areas of concern by describing how business objects relate to each other during collaborations. A business object represents a concept of interest within the organization, such as an activity or an actor. Business objects play multiple roles according to their behavior while collaborating with other business objects. Collaborations can then be expressed by the roles played by every participant in that scenario. This allows, on the one hand, creating semantically richer business process models, and, on the other, designing business objects where behavior is clearly separated and dependent on its usage context. Both of these results contribute in increasing the understandability of process models and business object reuse.

KEYWORDS

Business Process Modeling, Role Modeling, Object-Oriented Analysis and Design, UML, Organizational Engineering.

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 and relate with other organizational entities while supporting the operation of the business. The 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 (Madhavji 1991, Miers 1994, Eriksson 2001, Leymann 2002). Business process modeling can also be used for multiple purposes, such as facilitating human understanding and communication (Walford 1999), supporting process improvement and re-engineering through business process analysis and simulation (Eertink 1999, McGowan 1993), automating the execution of business processes (Aalst 2002, Scheer 1999) and facilitating coordinated business and system development by creating and keeping the alignment between business processes and their support systems (Chan 2002).

Modeling business processes involves capturing the structure of the multiple business objects, their relationships and collaborations. A business object represents a concept of interest in the organization, such as activities or human or automated actors. Identifying the business objects of an organization is fundamental to help documenting and evolving the business by facilitating communication and analysis. Additionally, if an organization holds a documented view on its business objects and their network of relationships this may foster business object reuse across different organizational process units.

However, properly identifying and modeling the business objects of an organization is not a straightforward task, especially when reuse is a concern. For instance, business objects are used by multiple

activities in different business contexts and processes. On the one hand, in a real organization, every business object relates to a set of multiple business objects. This leads to a highly connected relationship graph for every business object which may not be easy to document and understand. On the other hand, a business object exhibits different behavior according to the relationships it has at a given time. For example, a "product" resource plays different roles when relates to a financial activity or to a manufacturing activity. Yet, the specific context of a business object also defines it behavior. For example, a business object may behave as an activity when being performed or executed and as a business entity when being audited by another activity.

Despite these issues, identifying an organization's business objects so that their behavior is clearly specified and 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. To address these issues, this paper proposes a set of concepts, modeled as business objects (such as activities, entities and actors) where their relationship is specified by the roles the objects play in each collaboration. These concepts are modeled using object-oriented constructs and are represented as extensions to the Unified Modeling Language (UML). A role describes the behavior of an object in a specific collaboration and context, i.e., how an object is involved in a situation and what responsibilities it has. This allows creating a special perspective on business process, depicting the individual structure of business objects and describing business object collaborations according to the usage contexts of the involved objects.

The remaining of this paper is structured as follows: next section reviews related work on role modeling and its applications in software and business modeling. Section 3 introduces a role-based framework for business process modeling as an extension to the UML, along with some examples. Finally, section 4 concludes the paper and outlines future work directions.

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 a 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 context, Biddle and Thomas defined a role as "a collection of rights and duties relating to a position" (Biddle 1979).

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 software engineering that aims at separating multiple crosscutting concerns existing in a given domain. It is used in methodologies such as ISO's RM-ODP (1995) and especially in object-oriented frameworks (Gottlob 1996, Kristensen 1996, Kendall 1999, Halpin 2001). In this context, 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 can be abstracted as types, hence, there may be several instances of a role at a given time. Likewise, roles can also be instantiated, specialized and aggregated into composite models, promoting reuse. Roles emphasize on describing how objects interact with each other. While classes abstract common capabilities of individual objects, roles focus on the responsibilities of elements within a system, thus providing a different perspective on a class and its instances.

Class diagrams address information modeling but not interaction modeling. Role models identify the structure of elements and describe it as a structure of roles. Classes decompose objects based on their structural similarities and not because of their shared or collaborative activities and collaborations, whereas role models describing the same system in terms of their patterns of interaction. Therefore, roles provide an abstraction that is orthogonal to classes and objects, providing a complimentary view on object interaction. Role models do not replace classes or class models. Rather, they offer a specialized view on these elements.

In procedural and behavioral business 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 sub-activities (Curtis 1992). 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 (1995) introduced Role-Activity Diagrams to overcome this issue. 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 flow than sequential, such as parallel and conditional.

However, RADs do not fully depict activity context nor describe object relationships. First, activities are just procedural concepts and actors are the only concept subject to play a role (e.g. an information entity is not modeled according to the different roles it also plays). Second, an actor role is defined by grouping the full set of activities the actor can execute, thus describing its behavior. However, every non trivial actor executes different activities in different contexts. This is not captured and does not promote reuse and leads to identifying only macroscopic roles. Third, roles, as used in RADs, are not abstracted as types or classes, which holds back role specialization and reuse.

Recent approaches, such as Walford's (1999) and Eriksson's (2001), have explicitly integrated the object-oriented paradigm in business process modeling. The latter makes use of an UML subset 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). However, a role is used with the same meaning of that of RADs, representing the macroscopic responsibilities of actors or of organizational parties. Activities are descriptions of work that form one logical step within a business process. Activities are what organizational actors "do" in their roles.

3. ROLE-BASED BUSINESS PROCESS MODELING

Business objects are frequently described as isolated elements. While modeling or designing the concepts of interest in an organization as business objects, these are often described as independent units displaying a single and uniform set of capabilities and semantics. However, business objects interact and relate with other business objects and play roles for each other during these collaborations. For this reason, the semantics of a business object does depend on the information extracted from its relationships and collaborations. Therefore, these two perspectives are somewhat in conflict. The overall motivation for role-based business process modeling is allowing specific views on business objects. These views are used by other business objects as a means to improve the knowledge over the object's capabilities, allowing them to selective access the object. A view is a set of selected attributes and methods on an object. An important feature of these views is that they can change dynamically, i.e. be added or removed from a business object. Each one of these views is modeled as a role.

This section introduces a framework that makes use of roles to enhance object-oriented modeling of business processes. The framework is described as a specific profile to the UML using its extension mechanisms, as described next.

3.1 Extending the UML

The UML extensibility package specifies how UML model elements can be customized and extended with new semantics by the use of stereotypes, tagged values and constraints. A coherent set of such extensions defined for a specific purpose makes up a UML profile (OMG 2003, Alhir 1998). A stereotype is a user-defined meta-element that appears at the model layer of the UML four-layer meta-modeling hierarchy defining values, according to tagged values and constraints. Stereotypes allow new semantic meaning to be attached to a specific model element. Tagged values specify new properties or attributes that may be attached to an element. Finally, constraints are used to further refine the semantics of a new element by defining well-formedness rules and restrictions to one or more values or elements of the model. Constraints can be defined either informally or formally, for example, by using Object Constraint Language expressions (OMG 2003).

3.2 A Framework for Role-Based Business Process Modeling

An organization can be abstracted as a set of business objects, which are coordinated towards the achievement of 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 (Taylor 1995, Uschold 1998). This section summarizes a framework that focuses on describing business process modeling concepts along with their relationship with the supporting information system and information technology infrastructure. The object-oriented framework is described as an extension to the UML. For further details on this framework, we refer the reader to our work published elsewhere (Caetano 2003, Vasconcelos 2004).

Figure 1 depicts the stereotype declarations of the business domain concepts as specializations of the UML metaclass Class (in gray). As previously discussed, the top-level concept is that of business object, which is specialized as activity and entity. Actor and goal are entity specializations. Information system architecture stereotype declarations are not depicted.

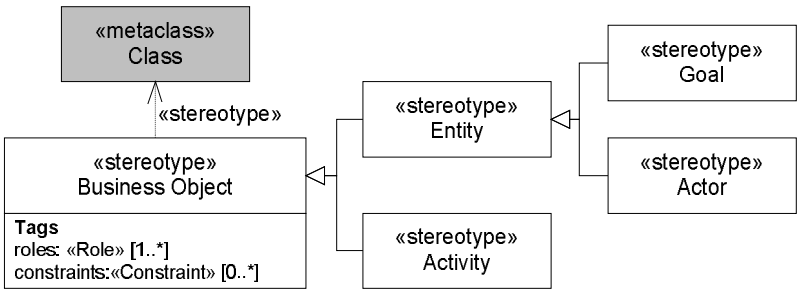

Figure 1. UML declaration of the business domain stereotypes.

Figure 2 is a class diagram depicting the relationships between the business concepts (in white) and information system architecture concepts (in gray). Activities, which can be functionally decomposed, relate to the goals they achieve, producing value to a customer. Activities are executed by at least one actor, during which they manipulate entities. To support their operation, activities make use of business services. These services are provided by information system blocks. Thus, automated actors are supported by information system blocks, which represent the functional components of the system architecture, including the mechanisms and operations required to manipulate information entities on the business domain. The organizational concepts that are relevant to keep information on are modeled as entities at business level and subsequently refined as information entities at information system level. Information technology blocks represent the application, platform and software components that implements information system block functionality. Each of these blocks provides technological services allowing block composition.

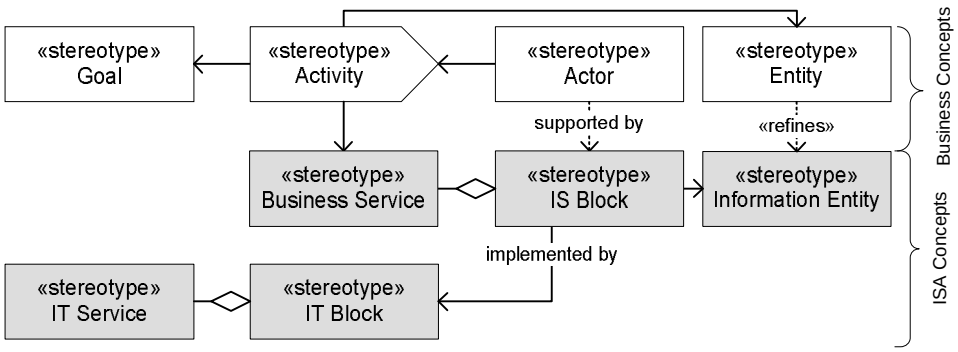

Figure 2. Relationships between the business domain stereotypes (elements in white) and information system architecture stereotypes (elements in grey).

3.3 Using Roles to Specify Business Object Collaborations

The core concepts behind the framework are that of business object and role. A *business object* represents a thing that is active in the business domain or organization. *Business objects* may collaborate with other *business objects*. Such collaborations are specified as relationships between one or more *roles* defined on a *role model*. A *role model* specifies role relationships and constraints, while a *role* defines the properties of a *business object* (i.e. its attributes and methods) that are relevant to be stated for it to behave as expected by other *business objects*. Thus, a role defines a unit of the object's observable behavior. A *constraint* asserts conditions over *business objects* or *roles*, specifying valid relationships. For example, a constraint may assert that entity E may not play roles R and S simultaneously or that activity A must be performed by actor M at least. Figure 3 and Table 1 summarize the UML stereotypes for representing these concepts.

Table 1. Description of role related stereotypes.

Stereotype	Base Class	Parent	Description
Role Collaboration Model	Model	–	A role collaboration model describes a role relationship pattern required for a set of business objects to carry out a valid collaboration. It consists of roles, the relationships between roles and constraints between these elements.
Role Collaboration System	Package	–	A role collaboration system is the top-level package in a role collaboration model and may contain roles, constraints and relationships.
Role	Class	–	A role defines a specific aspect of behavior.
Collaboration Constraint	Constraint	–	Asserts a condition over business objects and roles, thus limiting the extent of possible relationships between these elements. It may be expressed informally or formally.
Play	Association	–	An association between a business object (the role player) and a role. It is used to denote that the business object is able to achieve the behavior specified by that role.

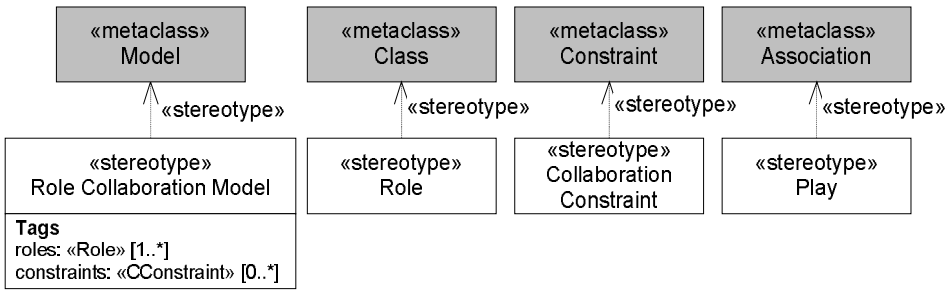

Figure 3. UML declaration of role related stereotypes.

Business objects are specialized to represent business domain concepts (please refer to Figure 1). These specializations are further described in Table 2.

An *activity* is a specialization of a *business object* and describes how to perform a piece of work. An activity corresponds to a verb in the business domain, which is performed by at least one *actor*, and operates over other *business objects*. We do not explicitly define the concept of business process in the framework since it can be structurally represented by functionally decomposing a top-level activity. A business process is an ordered set of activities that create a result with some value for a (external or internal) customer and contributes toward the achievement of goals.

An *entity* is a specialization of a *business object* and represents a passive concept, i.e. a business 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 of being created, accessed, modified, produced or consumed by the activity. 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). Three fundamental patterns of interaction between actors and activities can be observed: non-automated work (when all the actors assigned to an activity are human), semi-automated or facilitated work (when the work is

performed by an arrangement of human and mechanical actors), and automated work (when all the actors are mechanical). Finally, a *goal* depicts a measurable state the organization intends to achieve.

Table 2. Description of business domain stereotypes.

Stereotype	Base Class	Parent	Description
Business Object	Class	–	A modeled concept in the organization. Relationships between two or more business objects are realized by means of roles.
Activity	Class	Business Object	Describes how a piece of work is performed. Activities are performed by actors, and operate over business objects.
Entity	Class	Business Object	A passive class that depicts a concept of interest in the business domain. Entities usually participate in multiple business collaborations and outlive single interactions.
Actor	Class	Entity	Someone (human actor) or something (automated actor, such as an information system or a production machine) that can perform the actions required by an activity.
Goal	Class	Entity	A measurable state that the organization intends to achieve. Goals are achieved by performing activities.

To specify a business object one must represent it along with the roles it is able play, constraints and corresponding role model. In UML, this results in the diagram shown on the left of Figure 4. In order to simplify this diagram, we propose an alternate compact notation as shown on the right of the figure below. The compact notation will be used in the remaining of this paper.

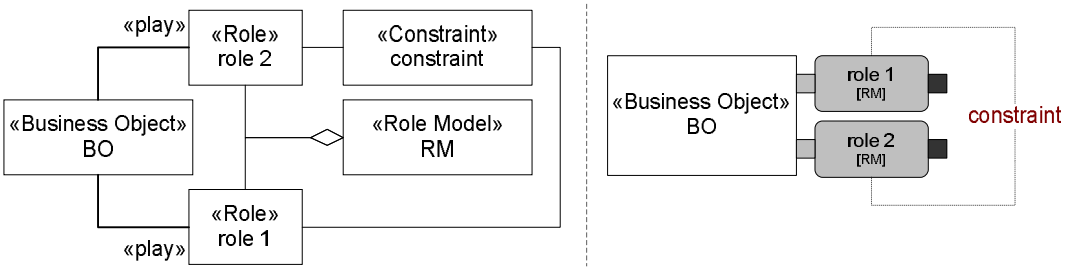

Figure 4. A business object, its roles, constraints and role models. UML notation (left). Compact notation (right).

3.4 Examples

This section illustrates the framework usage through two different simple 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.

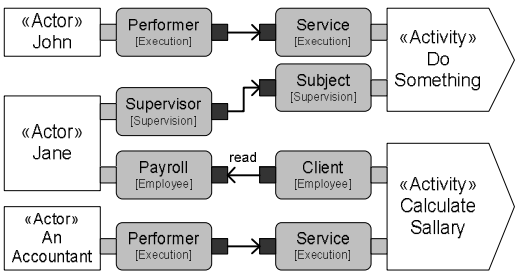

Figure 5. Actors and Activities

Figure 5 focuses on the relationships between three actors and two activities. Three role models (not shown on the above diagram) are used by the business objects in this example: Execution, Supervision and Employee. 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.” But

the same entity Jane is behaving as a resource when her payroll information is being “read” (read is a method from the Payroll role) 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.

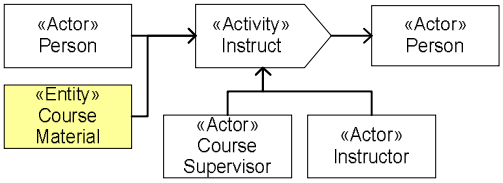

Figure 6. . Top-level process.

Figure 6 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).

However, the above diagram can be enhanced with roles so that the behavior of each business object is made clearer, as shown in Figure 7.

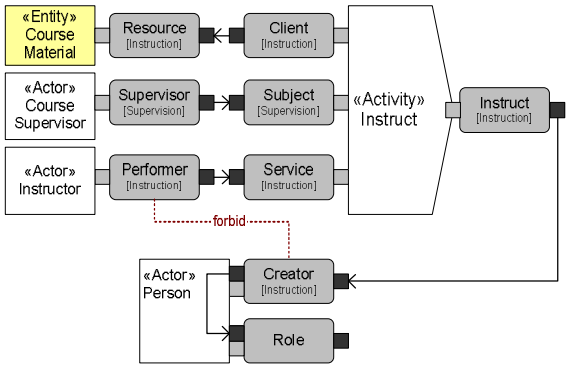

Figure 7. Top-level process with roles.

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 5, 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 can not be the same as the target of the instruction activity..

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 an extension to the UML meta-model using stereotypes. To illustrate the concept usage, two examples were shown which focused separating different concerns while modeling business objects. The proposed approach 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 within role models. The examples shown on this paper aim emphasizing that roles can be used to detail the collaboration patterns between business objects.

Concerning future research directions, we intend to address the specific collaboration patterns between actors and activities, by describing in a uniform way the services required by activities and the services

provided by actors, regardless of the type of actor. In the case of human actors, this requires modeling capabilities and skills with the goal of assisting actor-activity scheduling and scheduling assessment. In case of automated actors, requires contract and service binding modeling, aiming at facilitating the coordinated development of business processes and their supporting information systems and technology.

REFERENCES

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

Alhir, S. (1998, December). Unified Modeling Language Extension Mechanisms. *Distributed Computing*.

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

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

Chan Y. (2002). Why Haven't We Mastered Alignment? The Importance of the Informal Organization Structure. *MISQ Executive*, Vol. 1, No. 2.

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

Eertink, H. *et al.* (1999). A Business Process Design Language. *FM'99*, Vol. 1, LNCS 1708, Springer, pp. 76-95.

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

Gottlob G., Schrefl M., Röck B. (1996). Extending Object-Oriented Systems with Roles. *ACM Transactions on Information Systems*, 14. Page 268-296, July 3.

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. (1999). Agent Roles and Role Models: New Abstractions for Multiagent System Analysis and Design. *International Workshop on Intelligent Agents in Information Management*.

Kristiansen B. (1996). Object-Oriented Modeling with Roles. *Proceedings of the 1st International Conference on Object Information Systems*.

OMG (2003). *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.

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

Madhavji M. (1991). The Process Cycle. *Software Engineering Journal*, Vol. 6, No. 5.

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

Miers D. (1996). *Business Process Engineering*. Edited by C-T Colin, Kogan Page, London.

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

Taylor D. (1995). *Business Engineering with Object Technology*. John Wiley & Sons.

Uschold M., King M,, Moralee S., Zorgios Y. (1998). The Enterprise Ontology. *The Knowledge Engineering Review*, Vol. 13, Special Issue on Putting Ontologies to Use.

Vasconcelos A. Silva M., Fernandes, A. Tribolet. J (2004). An Information System Architectural Framework for Enterprise Application Integration. *Proceedings of the 37th Hawaii International Conference on System Sciences (HICS S37)*, IEEE Press, Hawaii, USA.

Walford T. (1999). *Business Process Implementation for IT Professionals and Managers*. Arthech House, Norwood, MA.