Winter 12-2-2007

A Conceptual Model for Negotiating in Service-Oriented Environments

Jyhjong Lin

Proceedings of The Seventh International Conference on Electronic Business, Taipei, Taiwan, December 2-6, 2007, pp. 419-424.

A CONCEPTUAL MODEL FOR NEGOTIATING IN SERVICE-ORIENTED ENVIRONMENTS

Jyhjong Lin, Ming Chuan University, Taiwan, jlin@mcu.edu.tw

ABSTRACT

Web services have been developed in recent years as a fundamental technique for the new generation of B2B or EAI applications. For their getting more available that imposes a new vision of service-oriented computing, the software industry has shifted its attention on software from developing a product as required to delivering a service on demand. In order to gain the many benefits from such a service-oriented model of software, several critical issues need to be addressed in a service-oriented environment such as differentiation of services with multiple attributes, dynamic selection and provision of services in a supply chain style, and commitment of services with prescribed rules. From the perspective of management, these issues are concerned within a process of negotiating desired services in a service-oriented environment. In this paper, we propose an object-oriented model that specifies such a negotiation process with explicit constructs addressing these critical issues. The model contains an architecture diagram that describes required components and their interactions for fulfilling the negotiation process, as well as a class/sequence diagram that specifies in detail what class objects these components have in order to collaboratively support all required behaviors occurred within the negotiation process.

Keywords: service-oriented model, negotiating, object-orientation, conceptual model

INTRODUCTION

For the rapid advances of Internet technologies in these years, Web services have been developed as a fundamental technique for the new generation of business-to-business (B2B) or enterprise application integration (EAI) applications. Until recently, with their underlying infrastructures such as XML [1,2], SOAP [3], UDDI [4], WSDL [5], WSCL [6], BPEL [7], and BPML [8] getting matured, more Web services have now become available that impose in the literature a new vision of service-oriented computing [9]. For the commonly recognized benefits by taking advantage of such a service-oriented vision [10,11], the software industry has shifted its attention on software from developing a product as required to delivering a service on demand in a service-oriented environment [12]. From the viewpoint of providing services, this means that software services are dynamically selected and delivered for tailoring the needs of an enterprise's business objectives; as one may also recognize, this implies as well many complex behaviors required for achieving the dynamic provision of services in order to deal with such a dynamic and changeable environment on the business/Internet nowadays.

Considerably, for these behaviors required for achieving the dynamic provision of services, their inherent complexity comes from the requirements that address several critical issues in a service-oriented environment. These issues in general include (1) differentiation of services from various providers with multiple attributes such as price, quality, and trust value; (2) dynamical selection and provision of services in a supply chain style where a composite service is possibly composed of a set of constituent services; (3) criticality of time for selecting services where a dynamic selection is required for each service request; (4) volatility of providing services where a service might not be available all the time; and (5) commitment of providing services with prescribed rules such as contract enactment and trust formulation. From the perspective of management, however, these critical issues can be seen as concerned within a process of negotiating desired services in a service-oriented environment [10,11]. It therefore becomes a major focus for all service participants (e.g., service requesters and providers) on imposing adequate mechanisms that support all required behaviors in a collaborative manner to accomplish such a negotiation process with explicit addressing on these critical issues.

Conceptual modeling is an important technique for representing a (part of) complex situation in an abstract manner with concise notations. It has been commonly used, for example, in analyzing and specifying user requirements of a computer-based application, as well as collecting and representing information required for dealing with complex technical and/or managerial issues to be resolved. Thus, to account for those aforementioned requirements, it is not uncommon to think of a conceptual model that specifies the negotiation process with respective constructs to support all required behaviors in dealing with those critical issues. In the literature, many technical discussions related to service-oriented computing and its corresponding negotiation process and considerable issues have already been presented as those in [9-15]. Nonetheless, any thorough conceptual models for specifying these requirements are still few nowadays; such models are indeed needed in that their specifications for these requirements are important in realizing a service-oriented environment – failure to specify these requirements usually results the environment in poor quality and high maintenance costs.

For these necessities, we propose in this paper a conceptual model for specifying the negotiation process with respective constructs to support all required behaviors. In general, conceptual modeling can be achieved by using function- [16-18], data- [19,20], or object-oriented [21-24] ways where the development of object-oriented ones is particularly motivated by the drawbacks and problems in the other two kinds: the significant features and benefits of object-oriented approaches would make resultant models more abstract and hence easier to be understood, maintained, and reused. As results, our model is object-oriented with UML [25-27] utilized as its modeling tool by taking advantage of such inherent features in the object-oriented paradigm as encapsulation of object specifics and interacted/coordinated nature of object behaviors to make it easier to be configured for an extensive support of specifying these requirements. Particularly, for the specification of the negotiation process, our model contains an architecture diagram that describes required components and their interactions for the fulfillment of the negotiation process. After then, for specifying required behaviors, our model uses class and sequence diagrams that present in detail what class objects these architectural components have in order to collaboratively support the behaviors occurred within the negotiation process.

This paper is organized as follows. Section 2 presents first the architecture diagram used in our model. The class and object sequence diagrams are respectively introduced then in sections 3 and 4. Finally, section 5 has the conclusions and future work.

THE ARCHITECTURE DIAGRAM

In a service-oriented environment, services are dynamically requested and delivered within a negotiation process. For the term negotiation is used in its sense to describe any negotiating among prospect participants for selecting desired services to be delivered by targeted providers. In our knowledge, the discussion in [11] for negotiating in a service-oriented environment presents a sound description about the negotiation process on top of a service-oriented architecture. Based on this idea, our model has first an architecture diagram that supports the negotiation process by imposing specific components required for fulfilling the considerable activities within the negotiation process. Figure 1 shows the most abstract view of our architecture diagram. In this diagram, three components are imposed where each one plays a designated role for participating in the negotiation process.

The service requester

The service requester is responsible for issuing service requests to the selected provider that delivers desired services after its negotiating for these requests with other negotiators (i.e., the service discovery agency and the service provider) has completed such that the service provider is discovered and selected with corresponding contracts signed and enacted. More specifically, for achieving its responsibilities, the requester takes usually the following issues into consideration:

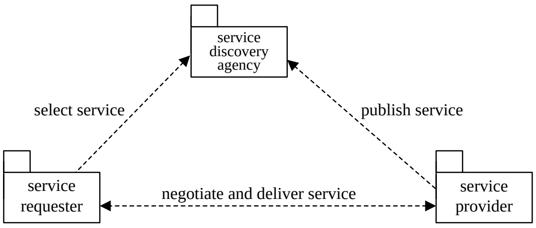

Figure 1: the architecture for negotiating in a service-oriented environment

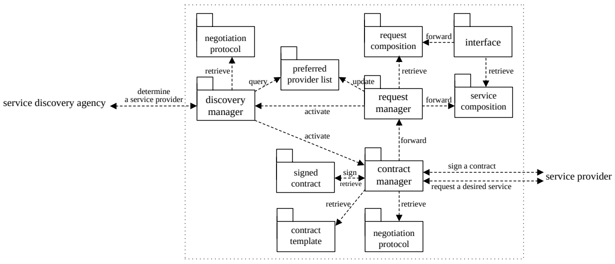

Figure 2: the service requester

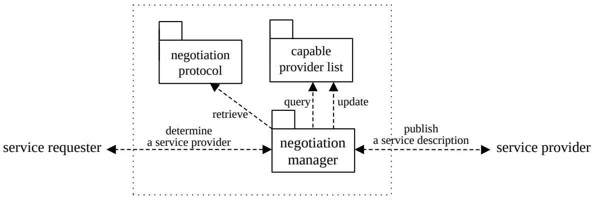

Figure 3: the service discovery agency

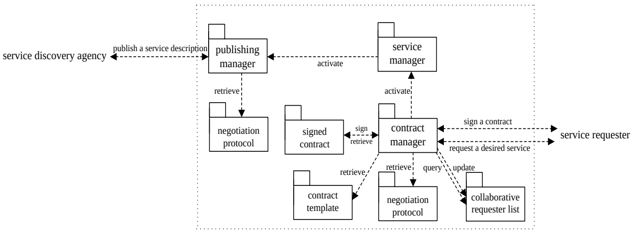

Figure 4: the service provider

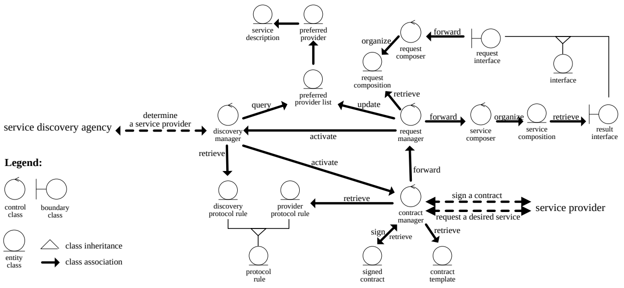

Figure 5: object class diagram for the service requester

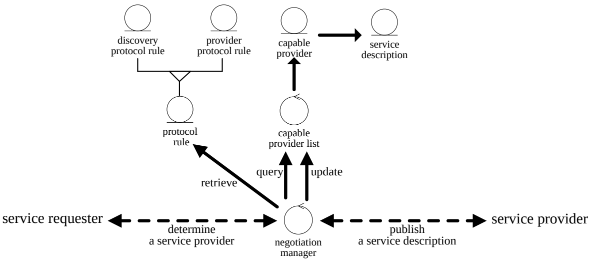

Figure 6: object class diagram for the service discovery agency

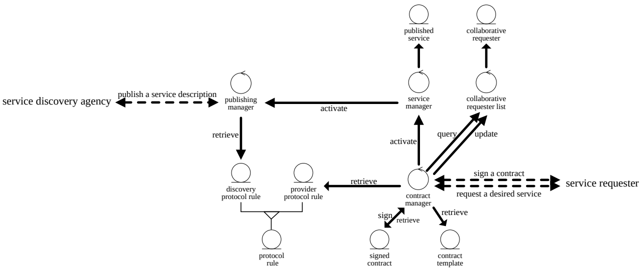

Figure 7: object class diagram for the service provider

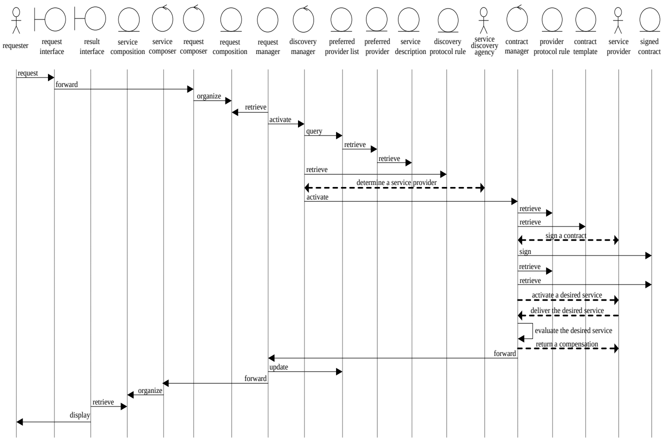

Figure 8: object sequence diagram for the service requester

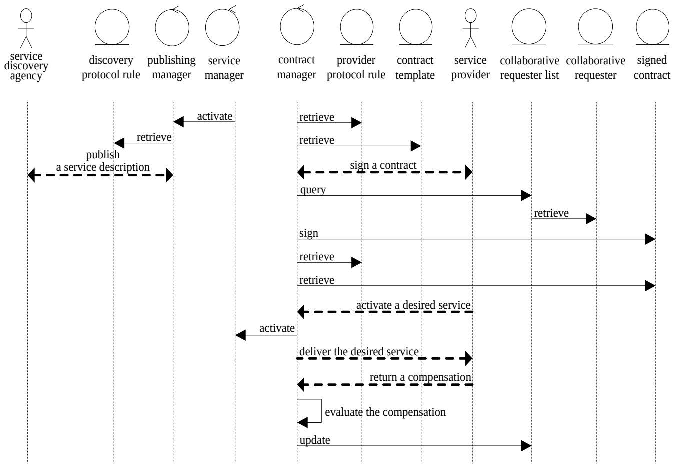

Figure 10: object sequence diagram for the service proivder

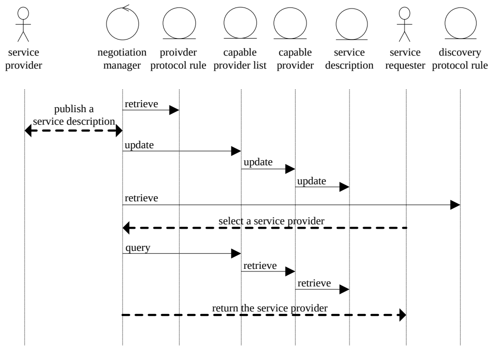

Figure 9: object sequence diagram for the service discovery agency

(a) for negotiations with the service discovery agency that determines a suitable service provider for satisfying a service request as well as with the selected service provider that responds the service request with delivering the desired service, a least protocol with associated rules to be complied among all negotiators is needed; therefore, the service requester must maintain an interaction protocol for proceeding the negotiations;

(b) since a negotiation encompasses a number of interactions among negotiators and hence be very time consuming, it is valuable for the service requester to adopt certain approaches that would be able to reduce as many as possible the number of invoked negotiations or to expedite as soon as possible the completion of invoked negotiations; as a common recognition, these approaches include usually (1) keeping a list of preferred providers and their providing services, and then when a request is necessarily issued, determining a selected provider directly and then reaching an agreement with the provider for signing and enacting a contract immediately; or (2) predicting the usage of delivered services based on the pattern of service requests, and then determining a selected provider and preparing a contract with the provider earlier before a request is necessarily issued (thus, the contract can be signed and enacted immediately while the request is necessarily issued);

(c) since determining a selected provider from a list of preferred providers would be often required for each request, it is valuable for the requester to adopt certain approaches that would be able to expedite as soon as possible the completion of the determination activity; a common approach is to structure the list in a sophisticated way (e.g., in a classified or indexed directory), and then expedite the determination by exploring the list via a convenient access (e.g., traversing the classified or indexed directory);

(d) service requests may be dependent among themselves due to their resulting from a composite request (e.g., a composite user requirement for achieving a business objective); in such a case, the service requester needs to maintain the relationship among these requests for negotiating and issuing these requests in an adequate sequence and also to deal with the possible consequences from negotiating or issuing these requests (e.g., success to discover and select service providers for these requests or failure to deliver services from these service providers);

(e) once negotiating for a request has completed and a service provider is discovered and selected, the service requester must sign a contract with the selected service provider; therefore, the service requester must maintain a contract template for completing the signing and enacting of the contract;

(f) once desired services are delivered under the promises denoted in corresponding contracts, the service requester should evaluate the effects of using these services (i.e., how these services are delivered in accordance with those promises) such that the trust values for the service provider can be updated for helping on the determination of selecting the same (preferred) service provider for further requests;

With the above issues concerned, Figure 2 shows the constituents in the service requester that collaborate to achieve its responsibilities.

The service discovery agency

The service discovery agency is responsible for determining a service provider that provides a service satisfying a desired service request from the service requester. As a common recognition, the determination is based on referring the desired service request to a set of service descriptions published by various service providers. Once that the desired requirements in the service request most suitably refer to a service description is reachable, the service discovery agency returns to the service requester the most suitable service description and its publishing provider. Similar to the service requester, for achieving its responsibilities, the discovery agency takes usually the following issues into consideration:

(a) for negotiations with the service requester for receiving its service request and determining a suitable service provider as well as with the service provider for accepting its publishing service description and being referred to a service request, a least protocol with associated rules to be complied among all negotiators is needed; therefore, the service discovery agency must maintain an interaction protocol for proceeding the negotiations;
(b) since determining a suitable service provider from a list of capable ones would be often required for each service request, it is valuable for the agency to adopt certain approaches that would be able to expedite as soon as possible the completion of the determination activity; a common approach is to structure the list of capable service providers that publish intended services in a sophisticated way (e.g., in a classified or indexed directory), and then expedite the determination by exploring the list via a convenient access (e.g., traversing the classified or indexed directory);

With the above issues concerned, Figure 3 shows the constituents in the service discovery agency that collaborate to achieve its responsibilities.

The service provider

The service provider is responsible for delivering its services to the desired service requester after its negotiating for these services with other negotiators (i.e., the service discovery agency for publishing these services and the service requester for delivering these services) has completed such that it is discovered and selected by the service requester with corresponding contracts signed and enacted. More specifically, for achieving its responsibilities, the provider takes usually the following issues into consideration:

(a) for negotiations with the service requester for receiving its service request and delivering the desired service as well as with the service discovery agency for publishing a service description to be referred to a desired service request, a least protocol with associated rules to be complied among all negotiators is needed; therefore, the service provider must maintain an interaction protocol for proceeding the negotiations;
(b) since delivering services to a service requester needs proper compensations, the trust values for the requester should be evaluated before delivery for ensuring its qualified collaborative role; therefore, the service provider must maintain a list of collaborative service requesters that possess trusted values for helping on the determination of delivering services to these requesters;
(c) once negotiating for a service request has completed, the service provider must sign a contract with the service requester for delivering the desired service; therefore, the service provider must maintain a contract template as in the service requester for completing the signing and enacting of the contract;
(d) once desired services are delivered under the promises denoted in corresponding contracts, the service provider should evaluate also the compensations for delivering these services (i.e., how these services are compensated in accordance with those promises) such that the trust values for the service requester can be updated for helping on the determination of delivering services to the service requester for its further requests;

With the above issues concerned, Figure 4 shows the constituents in the service provider that collaborate to achieve its responsibilities.

THE OBJECT CLASS DIAGRAM

Based on the architecture presented above, an object class diagram is then developed to describe what classes are required for defining objects allocated in architectural components to collaboratively support the behaviors occurred within the negotiation process. In UML, the ingredients in a class diagram can have three kinds of stereotype: boundary, entity, and control classes where a boundary class represents an interface used to interact the application with an actor as a bridge, an entity class models the information and associated behaviors in the real world, and a control class controls the access between interface and entity classes for accomplishing a desired behavior. Figure 5 shows our class diagram for the service requester based on the architectural diagram in Figure 2. It is noticed that as shown in this diagram, various relationships may occur between classes such as association and inheritance. As a common recognition for the object-oriented paradigm, these relationships (together with other features like information hiding in individual classes) are particularly useful for making the application constructed much easier to understand, maintain, and reuse. Figures 6 and 7 present the two class diagrams for the service discovery agency and the service provider based on the architectural diagrams in Figures 3 and 4 respectively.

THE OBJECT SEQUENCE DIAGRAM

With classes identified for creating objects in architectural components to collaboratively support the behaviors occurred within the negotiation process, it is now good time to create an object sequence diagram that specifies how such objects collaborate to realize these behaviors. Figure 8 is our sequence diagram for the service requester based on the class diagram in Figure 5, while Figures 9 and 10 present the two sequence diagrams for the service discovery agency and the service provider based on the class diagrams in Figures 6 and 7 respectively.

For illustration, as shown in Figure 8, the sequence of behaviors supported by objects in the service requester are: after a '*requester*' enters a (possibly composite) request via a '*request interface*', the '*request composer*' figures out first a '*request composition*' from the request in terms of a sequence of desired service requests to various service providers. This sequence of service requests are then retrieved by the '*request manager*' that activates in turn the '*discovery manager*' for determining service providers that provide services satisfying these requests. For each request, the '*discovery manager*' queries first the '*preferred provider list*' to ensure directly if a '*preferred provider*' can be found that provides a service (published via a '*service description*') satisfying the request. In the case that none of the preferred providers provides a service satisfying the request, the '*discovery manager*' negotiates with the third-party '*service discovery agency*' (under prescribed '*protocol rules*') for determining a suitable service provider. After recognizing a suitable service provider by the aid of the '*service discovery agency*', the '*discovery manager*' activates the '*contract manager*' that negotiates with the '*service provider*' (under prescribed '*protocol rules*') for the '*signing*' of a '*contract*' (under designated '*contract template*') to deliver the desired service under the commitment of the '*contract*'. After receiving the desired service, the '*contract manager*' evaluates the effects of using the service and then returns the received service together with the evaluated trust values for the '*service provider'* to the '*request manager*' that in turn updates the '*preferred provider list*' for helping on the determination of selecting the same (preferred) service provider for further requests. Finally, the '*request manager*' returns also the received service to the '*service composer*' that figures out a '*service composition*' to be displayed to the '*requester*' via a '*result interface*'.

CONCLUSIONS

Conceptual modeling is an important technique for representing complex situations in an abstract manner with concise notations. Motivated by the drawbacks in other methods, object-oriented modeling approaches are developed in order to result in a more natural, understandable, and maintainable representation. The method proposed in this paper is based on the most popular UML as its object- oriented modeling tool for an extensive support of the specification about the negotiation process in a service-oriented environment. In order to deal with the modeling complexity for the negotiation of desired services, components/constituents in each negotiator are identified and specified in a top-down fashion. As results, a higher-level architecture diagram is created first that describes required components with each one playing a designated role in the negotiation process. The detailed specification about what constituents these components have and how these constituents participate in various collaborations for achieving the role their containing component plays is then achieved by imposing step-by-step lower- level architecture, object class, and object sequence diagrams. We believe this provides a better way for understanding the negotiation process more naturally and easily. Finally, due to its formal semantics of the object sequence diagram, verification of supporting all required behaviors occurred within the negotiation process can be conducted via formal analysis of the diagram.

The work for negotiating desired services in a service oriented environment has already become a popular discussion. Although some technical researches about it have been done, none of them provides a thorough conceptual model for specifying the negotiation process with considerable issues concerned. In our knowledge, this would usually result the service oriented environment in poor quality and high maintenance costs. Our method presented herein provides an effort on this need by using object-oriented diagrams for specifying architectural components, their containing constituents, objects in these constituents, and how these objects collaborate to support the behaviors occurred in the negotiation process. We believe these diagrams presented in a step-by-step manner are much helpful for specifying those important requirements about requesting desired services and their achievements by negotiating in the service oriented environment.

As the technical issues about Web services are getting rapidly matured in these years, more Web services are expected to be available in the near future and hence a comprehensive mechanism for full supports of such a service-oriented computing environment will certainly become much more desirable. Thus, the development of such a mechanism is a desired field. In our view, using object-oriented techniques together with sound modeling constructs is a promising approach for an effective construction of the mechanism. In our future work, we will explore further some other key issues that our models have not addressed yet, including for an enterprise the recognition of its business objectives and how these objectives are specified and achieved by desired Web services under a committed service-oriented environment. As stated in [29], these issues should also be specifically addressed for keeping an enterprise competitive in a dynamic and changeable business environment on the Internet nowadays by recognizing and achieving its business objectives via the dynamic selection and provision of services in a committed service-oriented environment. Therefore, how to specify them by extending our modeling constructs will be carefully explored. Meanwhile, we will construct a tool to facilitate practical application of our models. These include a design environment for building the high-level architecture diagram and then deriving the detailed object class and sequence diagrams.

REFERENCES

[1] Extensible Markup Language (XML), http://www.w3.org/TR/xml11.

[2] Goldfarb, C. and Prescod, P. (1998), *The XML Handbook*, Prentice-Hall.

[3] Simple Object Access Protocol (SOAP), http://www.w3.org/2002/ws.

[4] Universal Discovery, Description, and Integration (UDDI), http://www.ibm.com/services/uddi/ standard.html.

[5] Web Services Description Language (WSDL), http://www.w3.org/TR/wsdl.

[6] Banerji, A., et al. (2002) "Web Services Conversion Language (WSCL) 1.0", *W3C note*.

[7] Andrews, T., et al. (2003) "Business Process Execution Language for Web Services (BPEL) 1.1".

[8] Business Process Modeling Language (BPML), http://www.bpmi.org.

[9] Champion, M., et al. (2003) "Web Service Architecture", *W3C Working Draft*,. http://www.w3c.org/ TR/2002/ WD-ws-arch-20021114.

[10] Elfatatry, A. (2002) "Service-Oriented Software: A Negotiation Process", *Ph.D. dissertation*, University of Manchester, Institute of Science and Technology, Manchester, U.K.

[11] Elfatatry, A. and Layzell, P. (2004) "Negotiating in Service-Oriented Environments", *CACM*, vol. 47, No. 8, pp. 103-108.

[12] Bennett, K., et al. (2000) "The Future for Flexible Software", *Proc. of the Asia-Pacific Software Engineering Conference, IEEE CS Press*, pp. 214- 222.

[13] Faratin, P. et al. (2000) "Using Similarity Criteria to Make Negotiation Trade-Offs", *Proc. of the 4th International Conference on Multi-Agent Systems*.

[14] Gottschalk, K., et al. (2002)"Introduction to Web Services Architecture", *IBM Journal*, vol. 41, No. 2.

[15] Jennings, N. (2000) "On Agent-Based Software Engineering", *Artificial Intelligence*, Vol. 117, pp. 277-296.

[16] Cameron, J. (1986) "An Overview of JSD", *IEEE Trans. on Software Engineering*, Vol. 12, No. 2, pp. 222-240.

[17] Jackson, M. (1983) *System Development*, Prentice-Hall.

[18] Yourdon, E. (1989) *Modern Structured Analysis*, Prentice-Hall.

[19] Hull, R. and King, R. (1987) "Semantic Data Modeling: Survey, Applications, and Research Issues", *ACM Computing Surveys*, Vol. 19, No. 3, pp. 201-260.

[20] Peckham, J. and Maryanski, F. (1988) "Semantic Data Models", *ACM Computing Surveys*, Vol. 20, No. 3, pp. 153-190.

[21] Hayes F. and Coleman, D. (1991) "Coherent Models for Object-Oriented Analysis", *ACM OOPSLA Conference*, pp. 171-183.

[22] Rumbaugh, J. , et al. (1991) *Object-Oriented Modeling and Design*, Prentice-Hall.

[23] Shlaer S. and Mellor, S. (1988) *Object-Oriented Systems Analysis*, Yourdon Press.

[24] Jacobson, I., Booch, G. and Rumbaugh, J. (1999) *The Unified Software Development Process*, Addison Wesley.

[25] Booch, G., et al. (2005) *The Unified Modeling Language User Guide*, Second Edition, Addison Wesley.

[26] Fowler, M. and Scott, K. (2000) *UML Distilled: Applying the Standard Object Modeling Language*, Second Edition, Addison Wesley.

[27] Rumbaugh, J., et al. (2004) *The Unified Modeling Language Reference Manual*, Second Edition, Addison Wesley.

[28] Quatrani, T. (2002) *Visual Modeling with Rational Rose 2002 and UML*, Addison Wesley.

[29] Casati, F., et al. (2003) "Business-Oriented Management of Web Services", *CACM*, vol. 46, pp. 55-60.