

INTRA-SERVICE ADAPTABILITY FOR ECA-CENTRIC WEB SERVICES USING CONTRACT AND ASPECT

Syed Saif ur Rahman*, Azeem Lodhi**, Nasreddine Aoumeur**, Claus Rautenstrauch**, Gunter Saake*

**Database Research Group, **Business Informatics Group, School of Computer Science, University of Magdeburg, Germany*

{srahman, mkhanlod, aoumeur, claus.rautenstrauch, saake}@iti.cs.unimagdeburg.de

ABSTRACT

In today's competitive environment, enterprise business has to adapt the changes driven by the market in order to achieve targeted goals. Information systems are used to manage the enterprise business, and thus adaptation at information system level is very important. It raises the need for architectures and mechanisms that support such adaptation at finer granularity (process activity level). Service-oriented computing and aspect-oriented programming approaches promises high adaptability in rule-based information systems. In this paper, authors propose an adaptable ECA (Event-Condition-Action) centric architecture with implementation mechanism, which is based on service-oriented computing and aspect-oriented programming concepts to adapt changes at intra-activity level for rule-based enterprise information systems.

KEYWORDS

Rule-based adaptation, Aspect-oriented software development, Contract, Workflows, Business rules, Workflow Engines.

1. INTRODUCTION

Business world is competitive and changing continuously, this requires the changes to be adapted in enterprises to keep strategic position in the market. Enterprises are governed by business processes, which can be described as "a structured, measured set of activities designed to produce a specified output for a particular customer or market" [B. Francois et al, 2006]. Business process consists of activities/operations which are executed in a particular order as defined by business rules. According to the Business Rules Group¹, a business rule is a statement that defines or constrains some aspect of the business. Business rules are intended to assert business structure or to control the behavior of the businesses in execution. In real-time business environment, changes driven by the market are needed to be incorporated at activities level in form of rules, designed according to business needs; to keep businesses running with best possible results.

Enterprise businesses are managed by enterprise business information systems. The integration of enterprise business with information systems raises the need for adaptation of changes to be performed at information system level concurrently as they occur in enterprise business. Such adaptation in information systems is incorporated as workflows [Workflow Management Coalition, 1999], [F. Leymann and D. Roller, 2000], [G. Dimitrios et al, 1995], or rules that surround executing business processes and activities.

Legacy information systems were developed using tightly-coupled architectures and were hard to upgrade or adapt as per changing business requirements. This resulted in creation of loosely-coupled open system architectures and platforms supporting high interoperability and adaptability. Advent of web services [G. Alonso, et al, 2004], [P. Cauldwell et al. 2001], [E. Cerami, 2002], service-oriented computing [M. P. Papazoglou, 2003], workflow systems [Workflow Management Coalition, 1999], [F. Leymann and D. Roller, 2000], [G. Dimitrios et al, 1995], and rules execution engines enabled high interoperability and adaptability for business processes at information system level.

¹ Business Rules Group, 2000. Business Rules Group, July 2000.

Business processes are subjected to change because of market factors. These changes may drive enterprises to provide new services or to extend the existing services by collaborating with other partners (enterprises). In order to cooperate with other enterprise services, rules and conditions may have to be changed to incorporate other enterprise requirements. For providing new services or modifying the existing functionalities (services), Aspect Oriented Programming (AOP) [G. Kiczales et al, 1997] can be used. In this paper, authors proposed an adaptable ECA (Event-Condition-Action) centric architecture and implementation mechanism based on service-oriented computing and aspect-oriented programming concepts to adapt changes at intra-activity level for rule-based enterprise information systems. Aspect Oriented Programming (AOP) can be used for providing new services or modifying the existing functionalities (services). In proposed solution AOP is used to achieve adaptability at intra-activity level. Benefit of using this technique is the separation of cross-cutting and optional rules-based-operations as aspects and possibility of weaving these aspects at runtime to achieve high re-configurability. For explanation and implementation of the approach, a bona fide case-study is discussed in banking system which shows the rule-based collaboration between customer and account (i.e. bank) for cash with-draw activity.

In section 2, we will give the motivation for proposed architecture. Section 3 provides the conceptual background followed by section 4 which discuss the involved technologies for implementation of the approach. Section 5 discusses the proposed approach and case study. Section 6 discusses related work whereas conclusion and future work is mentioned in section 7.

2. MOTIVATION

Business processes are continuously subject to change over time because of changing requirements. Adapting the requirements and business changes requires new services or extension of existing services. To incorporate these requirements in enterprise; services, rules and conditions may have to be changed. Enterprise business information systems execute the business workflows based on rules. In classical information systems, business rules and the execution of operations/actions based on these rules are hard coded, that hinders the adaptability of the system. Workflows and rules can be within single activity or across several activities; we term workflows and rules with in single activity as intra-activity (IA) workflows and rules. Each rule is based on ECA (Event-Condition-Action) specification, which offers a flexible, adaptive, and modular approach for realizing business processes [B. Francois et al, 2006]. If ECA rules are fixed then adaptation to changes in business at information system level is difficult. We recommend adaptable ECA rules based on contract between participating partners and rule-based aspectual weaving of operations at intra-activity level for developing highly adaptable enterprise business information systems.

3. CONCEPTUAL BACKGROUND

In SOA environment, business processes are constructed in terms of services, which are loosely coupled and execute certain activities/services according to set of rules to fulfill the business goal. Thus services may adapt the changes without having effect on other services. Services are executed in response of an event, which occurs in the business environment. When event occurs, corresponding business rules (policies, conditions) are checked and then actions are taken to give response and to execute business process. Business rules (policies, conditions) decide how the processes (activities) will be executed based on the involved attributes of entities.

Business processes have to adapt the changes as driven by the market, these changes need to be reflected in information system as they manage the business processes. In information systems, business processes (& activities in them) are executed based on workflows. Workflow is the sequence of operations, activities that are executed in some predictable order as per defined rules (conditions). These rules are correspondingly implemented in IS as workflow rules. Rules in workflow can be defined like if-else statements in programming languages or more precisely in event-condition-action (ECA) format using declarative approach. Contract is defined as a relationship between participating partners within business process or activities. Contracts are instantiated during the execution of workflows based on the participating partners and the triggered rules. Contracts can be instantiated or canceled based on the business relationships. The

relationship between business process, activities, rules, contracts and participating partners is depicted in figure 1.

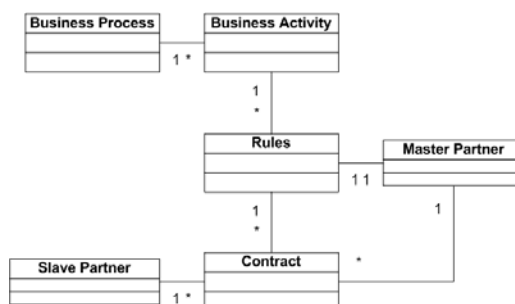


Figure 1: Domain-entities relationship diagram.

As per the figure, each business process is consisting of one or more business activities. Business activities are executed according to one or more rules. Rules involved in execution of activity are mapped with participating partners with the help of contracts. Two kinds of participating partners are defined i.e., master partner and slave partner. Master partner is directly mapped to the rule and it is responsible for identification of rule for execution based on its characteristics. Slave partner is offered contract based on rule and once accepted; it is obliged to follow the contract. Each partner may have one or more contracts for single rule.

AOP is methodology emerged with the aim to separate cross cutting concerns. It prevents code tangling and scattering, ensuring code scalability and maintenance. Using AOP, cross-cutting code is separated from the program logic using Aspects. Using join-points, point-cuts and advice information, aspect weaver brings the program code and aspect together. Join points are events of interest for aspect weaving, point-cut are the collection of join points and advice is the intended behavior to be weaved.

Adaptability means adaptation of changes to the system. Information systems need adaptability at different levels of granularity, from finer grained (i.e. intra-activity level) to coarser grained (i.e. business process level). We can also categorize adaptability mechanism in information systems based on the categories like specification/description, discovery, composition, execution, management etc. For example, we can make web service adaptable in term of service description i.e. web service description can be adapted based on the service user (i.e. specification/description category) or we can also adapt service execution based on the service consumer (i.e. execution category). In this work authors focus fine grained adaptability at intra-activity level and we can categorize it as activity execution level adaptability.

4. INVOLVED TECHNOLOGIES

Web service is implementation technique for SOA and standard defined by W3C to support interoperability between interacting systems over internet. As per concepts defined in W3C standards, systems interacting using web service communicate via XML messages based on SOAP² (Simple Object Access Protocol) standard, so web service is based on message passing interactions. Interacting systems expose the functionalities using the WSDL³ (Web Service Description Language) as per W3C recommendation. Web service possesses good potential in business process domain because of simplicity and interoperability. Other benefits that web services possess include end-point platform independence and automatic discovery and usage. We recommend the use of web services for implementing business processes and activities to ensure that services will be available to any client using standard protocols.

.NET framework is Microsoft proprietary managed code programming model for application development. It supports syntax for multiple programming languages including C#, VB.NET, J# and C++ all generating Common Intermediate Language (CIL) on compilation. .NET platform provides excellent support

² Simple object access protocol (v.1.1), 2000. The World Wide Web Consortium (W3C). Published on Web (May 2000).

³ Web service definition language (v.1.1), 2001. The World Wide Web Consortium (W3C). Published on Web (Mar. 2001).

for software extensibility, adaptability, maintainability and customizability through various techniques including reflection, proxy, intermediate language and on-fly code generation.

Windows Workflow Foundation (WF) is Microsoft proprietary programming model to develop and deploy workflows in applications. It supports authoring, executing and managing workflows. WF supports two types for workflows i.e., Sequential Workflow and State-Machine Workflow. Sequential workflow executes in sequential order however state-machine workflows execute based on the set of states. WF consists of an activity model, workflow designer, workflow runtime and rules engine. WF stores rules in XML format.

Spring .NET application framework⁴ is an open source enterprise .NET applications development framework, developed and maintained by SpringSource⁵. It provides various tools for enterprise .NET application development including Spring.Aop, a framework for Aspect-Oriented Programming (AOP) support in .NET. Spring .NET AOP framework support dynamic weaving at runtime and is developed using Microsoft .NET C#. For AOP, Spring .NET utilizes interception of method invocation mechanism. Spring .NET AOP uses Spring .NET IoC (Inversion of Control) container internally for managing advice and point-cuts. Advice can be defined using object definition syntax. Spring .NET at runtime generates AOP proxies using .NET reflection capabilities.

5. IMPLEMENTATION OF APPROACH & CASE STUDY

In this paper, business activities are defined using intra-activity workflows. Services are used to expose activity interfaces among participating partners. Each service executes the workflow for activity as per defined contract and rules. The proposed approach is to utilize rule-based-contract between participating partners for adaptability. This rule-based-contract is applied at intra-activity level of business processes to ensure the adaptability at fine grained level. Contract means the agreement between participating partners based on related attributes during activity/operation execution. The proposed approach recommends to instantiate contract between participating partners based on executing rule i.e., for each rule different contracts can be defined, which ensures adaptability of contracts at fine granularity. Aspectual adaptation can be done at the rule execution level to add/remove cross-cutting and/or optional functionalities. If needed each rule execution may involve the weaving as aspects of crosscutting or optional functionality and instantiation of contract from database. This approach ensures that changes at intra-activity level are adaptable. Compliant mapping of ECA to .NET implementation is shown in figure 2.

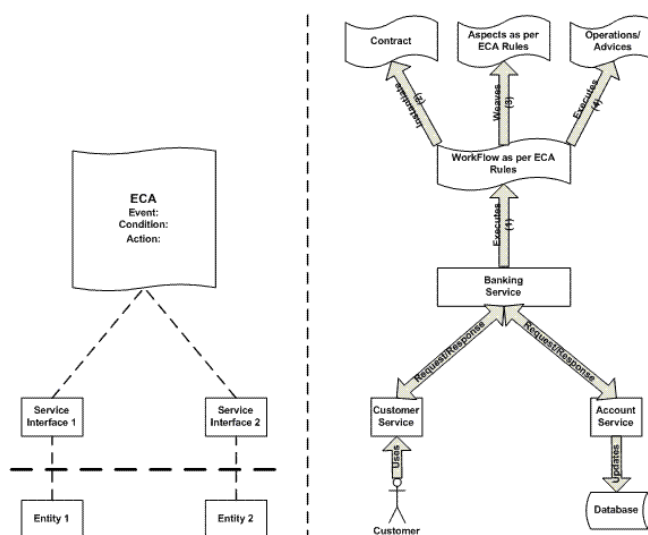


Figure 2: Compliant mapping of ECA to .NET/J2EE.

⁴ SPRING .net Application Framework, <http://www.springframework.net/>.

⁵ SpringSource, <http://www.springsource.com/web/guest/home>.

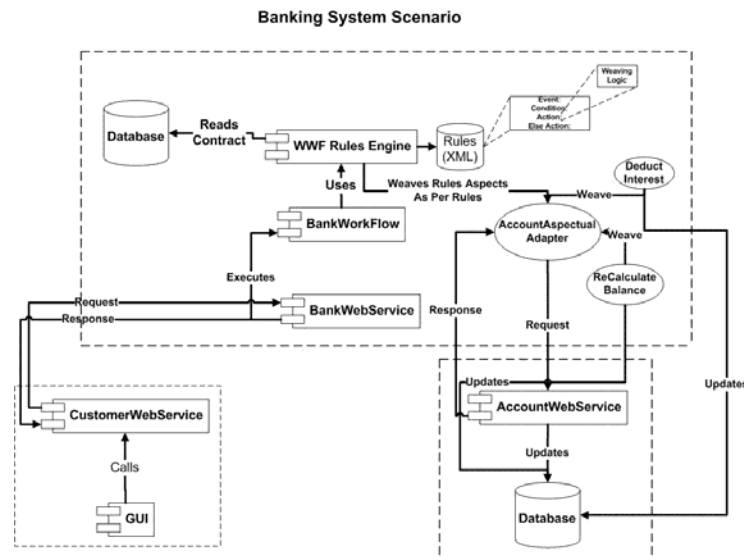


Figure 3: Compliant mapping of ECA to .NET.

In this implementation, term service means a web service implemented in .NET and is used for deploying business processes and activities. Workflows are implemented using Microsoft Workflow Foundation (WF). We used aspect-oriented programming (AOP) to weave operations-as-per-rules as aspect into intra-activity workflow execution. This implementation has separated cross-cutting and optional rules-based-operations as aspects. Weaving details of these aspects is defined in rules. This ensures that such rules are not hard coded and can be changed as needed. It also ensures that operations that are not needed by activity as per rule are not loaded ensuring reduced complexity and size of instantiated object. Rule execution engine weaves the rules-based-operations as aspects at runtime. Since cross-cutting operations are separated as aspects, system implementation is much cleaner resulting in better system maintenance and upgrade. The architecture for implementation is demonstrated in figure 3.

The proposed approach and implementation can be best explained with the help of a case study. The case study represents a banking environment, where business activity "with-draw money" is executed. A customer (i.e. slave partner) can own one or more bank accounts (i.e. master partner) with different account categories (i.e. master partner characteristics). A customer may have different contracts with account (i.e. bank) for credit limit and interest rate based on the rule. From customer point of view, whenever he/she withdraws cash from bank the credit limit and interest rate is applied, and should be precisely reported to the end user. From Bank point of view, based on the category of account, balance of customer and the amount requested for withdrawal, system should automatically calculate credit limit and interest rate as per business rule and contract.

The proposed and implemented mechanism to achieve the adaptability at the intra-activity level is described below as step-wise description:

1. At the first step, business process is defined. Since business processes are composed of multiple activities, so business activities are defined and mapping of business activities with processes is carried out. Each activity performs set of operations based on defined workflow and rules; we refer to this workflow as intra-activity (IA) workflow. Workflow and rules for activity is defined using work flow tool. The rules can be modified/adapted any time and will identify the interacting partners and their attributes for contract.
2. Till now, we have defined everything as per traditional standard implementations. However to achieve adaptability the association between partners for activity execution can be defined or updated at runtime using screen. The rule-centric contract-based approach should be realized at this level. We will define contract between partner entities based on governing IA rules. This contract will contain the definition of common attributes for participating partners on rule. Once intra-activity rule-centric contracts between partner entities are negotiated, and then operations can be executed based on instantiated contract. Figure 4 shows the "IA Rule Instantiation" screen of the implementation.

- The original objective of this implementation is to execute business activity in adaptable manner. This approach provides transparent activity execution mechanism i.e. system automatically executes activities based on the defined contracts between the participating partners of the activity. Screen for execution of activity based on invoked rules is shown in figure 5.
- Now as the business evolves, changes are needed to adapt the business requirements and needs. This triggers the change in intra-activity rules. As we separated the rules logic from participating partners in terms of contract; adaptation to changes is achievable through rule and contract update.
- In traditional systems operations/actions executed based on rules were hard coded into rule definition, even such systems evolved to support dynamic operations/actions execution as per rule but these operations/actions became the part of the executing activity at both design and execution level. This resulted in large complex source code with less modularity and maintainability at design time. The objects were instantiated with functionalities that were never used. We proposed the solution to this problem using aspect-oriented programming technique for operation/action integration in activity execution based on intra-activity rule definition. Rule based operations/actions should be separated as aspect. As per rule definition these aspects are weaved into activity based on workflow and rule execution resulting in high modularity, maintainability and low complexity. Figure 6 show aspectual adaptation mechanism.

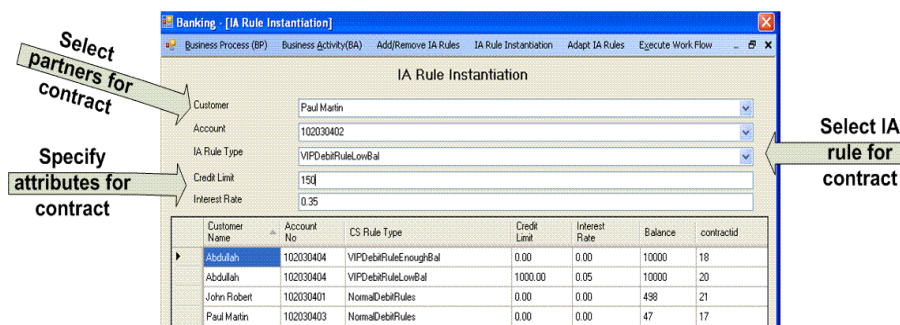


Figure 4: IA rule instantiation screen.

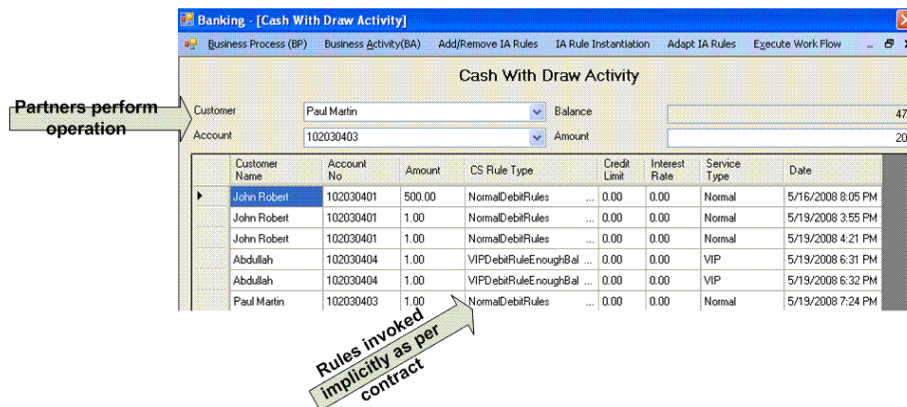


Figure 5: Cash with draw activity screen.

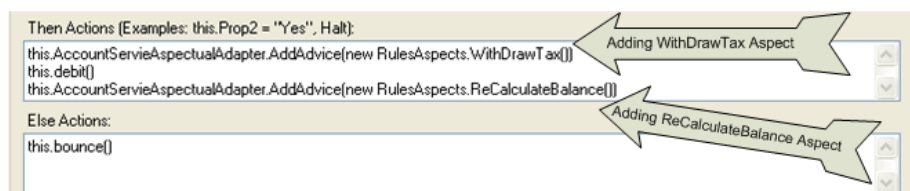


Figure 6: Aspectual adaptation.

In this case-study we implemented one workflow for cash with-draw activity. The intent for this implementation is to achieve adaptability at intra-activity level. As per workflow the very first step is instantiating the rule contract. Once branch condition are satisfied during workflow execution, it is finalized which rule will be executed for activity. As per the rule the contract is instantiated and activity operations are executed based on instantiated contract for triggered rule.

6. RELATED WORK

Adaptability is an active research area in the domain of business processes. Event-Condition-Action (ECA) rules offer a flexible, adaptive, and modular approach for realizing business processes [B. Francois et al, 2006]. Business processes should be dynamic, and web services based on rules based approach supports realization of dynamic business processes [F. Curbera et al, 2002]. The potentials of business rules in web services have firstly been explored in [B. Orriëns et al, 2003]. A stepwise rule-driven methodology is proposed to enhance the dynamic adaptability while composing web services using BPEL-like standards. The milestone consists in judiciously classifying in a repository, different business rules specific to web services composition. Nevertheless, no formal verification / validation of the constructed composition are undertaken. Another proposal [F. Rosenberg and S. Dustdar, 2005] conceives business rules as web services described using extensions to reactive RuleML, instead of the passive and static WSDL. The approach is automated with supporting tool called ViDre. Nonetheless, the approach does not tackle the conceptualization level neither copes with dynamic composition of the (WS) rules to specific business process activities. A. Charfi et al. [A. Charfi and M. Mezini, 2007] leverages BPEL-like with more agility and modularity, by enhancing it with an extra aspectual level. The resulting new language named AO4BPEL, allows thus externalizing crosscutting concerns such as security and data handling as advices. AO4BPEL uses XMLQuery language XPath as its pointcuts language. The approach to agile services introduced by Erradi et al. [A. Erradi et al, 2006] also adopts aspect-orientation, and is specifically devoted to policies and QoS concerns. In the same line, A.Finkelstein et al. proposed in [C. Courbis and A. Finkelstein, 2005] a generic aspect-oriented language; they applied for dynamically weaving behavioral advices on BPEL code. Another aspect-driven approach to web services appeared most recently in [S. Ruy et al, 2007]. Most of the existing work in the field is related to adaptability at service composition level; however this work is unique in a way that it realizes adaptation at intra-activity level. Contract based approach is used to ensure each partner have adaptability for each instance. Aspect-oriented adaptation ensures that operations can be weaved as per ECA rule-actions as needed and can be used to alter the weaving details using rule adaptation approach.

7. CONCLUSION AND FUTURE WORK

The proposed approach and implementation in this paper ensures that rules are adaptable during the complete lifecycle of the information system. Contract based approach is used so that rule execution is adaptable at the instance level characteristics of participating partners. Aspect-oriented approach ensures the adaptability at ECA action level through weaving and un-weaving of operational aspects as per the rule execution. For future work we intend to extend the adaptability at inter-activity level and business process level specially using BPEL⁶.

ACKNOWLEDGEMENT

We are grateful to the reviewers for helping in improving this final version. This work is partially funded by Higher Education Commission of Pakistan.

⁶ OASIS. Web service business process execution language 2.0, 2006. Last accessed: Aug. 11, 2008.

REFERENCES

- A. Charfi and M. Mezini, 2007. AO4BPEL: An Aspect-oriented Extension to BPEL. *World Wide Web Journal: Recent Advances on Web Services (special issue)*, 10:309–344, 2007.
- A. Erradi et al, 2006. Policy-Driven Middleware for Self-adaptation of Web Services Compositions. *In ACM/IFIP/USENIX 7th International Middleware Conference, volume 4290 of Lecture Notes in Computer Science*, pages 62–80. Springer, 2006.
- B. Francois et al, 2006. Realizing business processes with ECA rules: Benefits, challenges, limits. *In PPSWR*, pages 48–62, 2006.
- B. Orriëns et al, 2003. A Framework for Business Rule Driven Web Service Composition. *In Proc. Of Conceptual Modeling for Novel Application Domains, volume 2814 of Lecture Notes in Computer Science*, pages 52–64. Springer, 2003.
- C. Courbis and A. Finkelstein, 2005. Towards aspect weaving applications. *In Proceedings of the 27th international conference on Software engineering (ICSE '05)*, pages 69–77. ACM Press, 2005.
- E. Cerami, 2002. Web Services Essentials - Distributed Applications with XML-RPC SOAP UDDI & WSDL. O'Reilly, Feb 2002.
- F. Curbera et al, 2002. Business process execution language for web services, July 31 2002.
- F. Leymann and D. Roller, 2000. Production Workflow. Prentice Hall, 2000.
- F. Rosenberg and S. Dustdar, 2005. Towards a Distributed Service-Oriented Business Rules System. *In Proc. of the IEEE European Conference on Web services (ECOWS)*. IEEE Computer Society Press, 2005.
- G. Dimitrios et al, 1995. An overview of workflow management: From process modeling to workflow automation infrastructure. *In Distributed and Parallel Databases*, page 119-153, 1995.
- G. Alonso, et al, 2004. Web Services: Concepts Architectures and Applications. Springer, 2004.
- G. Kiczales et al, 1997. Aspect-oriented programming. In M. Akit and S. Matsuoka, editors, *In ECOOP'97 Object-Oriented Programming, 11th European Conf., volume 1241 of Eds. LNCS, Springer-Verlag, Berlin*, page 220-242, 1997.
- M. P. Papazoglou, 2003. Service-oriented computing: Concepts, characteristics and directions. *In 4th Int. Conference on Web Information Systems Engineering (WISE'03)*, 2003.
- P. Cauldwell et al. 2001. Professional XML Web Services. Wrox Press, Birmingham, USA, 2001.
- S. Ruy et al, 2007. A Framework for Managing the Evolution of Business Protocols in Web Services. *In Asia-Pacific Conference on Conceptual Modeling (APCCM'07)*, 2007.
- Workflow Management Coalition, 1999. Workflow Management Coalition. Process Definition Interchange Process Model. *Published on the World Wide Web by the Workflow Management Coalition as Document Number WfMC TC-1016-P*, Version 1.1, 1999.