Inter-Organizational Information System Architecture: A Service-Oriented Approach

Mohammad Kazem Haki and Maia Wentland Forte

Faculty of Business and Economics, Institute of Information Systems,
University of Lausanne, Switzerland
`{kazem.haki,mwf}@unil.ch`

Abstract. The strategic value of Inter-Organizational Information System (IOIS), regarded as an automated information system shared by two or more companies, highly depends on its architecture to support agile business needs. The main requirements of this architecture for supporting business agility are *flexibility and scalability*, *interoperability and integration* as well as *process-centric architecture and continuous business process improvement*. We claim that the service-oriented architecture (SOA) approach, the latest in a long series of attempts in software engineering, is the best for meeting these requirements. In this paper, after investigating the compatibility of SOA features with IOISs requirements, we propose a service-oriented IOIS architecture composed of business service, application and integration layers.

Keywords: Inter-Organizational Information Systems (IOISs), Service-Oriented Architecture (SOA), Business Agility.

1 Introduction

Inter-organizational systems are information and communication technology-based systems that deal with Business-to-Business supply chain interactions. Transcending legal enterprise boundaries [16], they are considered as one of the major critical success factors for getting a real time response to the continuous changes of the market. When mobilized properly, they can bring the organization a significant competitive advantage. The role IT plays in these systems, called Inter-Organizational Information Systems (IOISs), is strategic. Hence, the introduction of new IT technologies opens new architectural avenues. IT is currently moving from object-orientation to service-oriented computing, now considering services as fundamental elements for application development [27]. Thus Service-Oriented Architecture (SOA) is becoming the fundamental approach for developing the next generation of IOISs. Regardless of the trend, compatibility between the SOA specifications and the IOISs requirements leads to service-oriented IOISs.

   In this paper we will propose a service oriented IOISs architecture that provides a solution meeting the requirements of IOISs. After examining these requirements, based on the literature, we will propose a classification highlighting the potential of using SOA. We will then describe the characteristics of SOA and prove the compatibility of

its capabilities and features with IOIS requirements. Based on this demonstration, we will conclude by proposing a top-down three layers service-oriented architecture for IOISs.

2   Requirements of IOISs

Inter-organizational information systems (IOISs) date back to 1966, when Kaufman raised awareness on information systems that operated across organizational boundaries [13]. IOISs, defined by [5] as "an automated information system shared by two or more companies", aims at facilitating the creation, storage, transformation and transmission of information that transcends legal enterprise boundaries [12, 16]. Indeed the use, by two or more organizations, of a one and same system is considered necessary and sufficient to qualify it as inter-organizational [7].

An IOIS differs from an internal, distributed information system in that it allows information to be sent across organizational boundaries. Access to stored data and applications programs is obviously shared to varying degrees depending on the participants' specific contexts [12]. When compared to internal IS, IOISs following characteristics have been identified by [29]: 1) they require partners which are willing, able and ready to cooperate 2) standards play a key role (e.g. data communications protocols, company policies), 3) third parties are often involved such as software-service providers, partners for supporting inter-organizational business processes, etc., 4) the work must be synchronized among partners 5) technical aspects are less important than the new electronic relationships involved i.e. the importance of electronic business models and 6) efforts cannot be secretive, i.e. IOIS requires more openness especially when industry standards are adopted.

Looked at from a operational point of view, IOISs lead to real-time interaction, higher transaction accuracy, more efficient and quicker payments, rapid response, reduced search costs, reduction in inventory and tighter links to customers. These benefits enable all parties to have high operational efficiency and capability, explaining why more and more corporations tend to adopt IOIS in order to gain competitive advantages [12, 21].

In the literature of IOISs different buzzwords have appeared reflecting the main characteristics of IOISs. Among those, one can cite extranet, EDI (Electronic Data Exchange), Internet EDI, B2B e-commerce, e-SCM (Supply Chain Management), etc. In effect, IOISs covers all the spectrum from simple EDI systems for exchanging data such as purchase orders to more complex transactions such as shared technical databases. Depending on the characteristics of each IOIS, the implementation needs to take into account some specifics. Hereafter, we mention the result of some studies in this regard we consider as pertinent for our purpose.

Critical Success Factors (CSF) of IOISs:
 [21] mentioned seven CSFs regrouped within three clusters:
   1. Decision motivation: 1) strong motivation is essential to gain and thereby guarantee partners' commitment 2) shared motivation and vision is essential;
   2. Implementation process: 3) cross-organizational implementation team is necessary 4) high integration with internal information systems is indispensable 5) inter-organizational business process re-engineering could be needed;

3. Infrastructure conditions: 6) advanced legacy information system and infrastructure should be taken care of 7) shared industry standard should exist.

These CSFs show the technical and non-technical requirements for implementing IOISs.

IOISs requirements from an integration perspective:
On one hand, integration contributes to exchanging data so as to unify software components and streamline business processes. On the other hand, tight integration of IS in different companies may well reduce flexibility and agility of these organizations.

[30] studied the integration requirements of an IOIS from a two dimensions perspective determining some criteria for each, as follows:
1. Intensity of integration: scope of integration, type of integration
2. Flexibility of integration: scope of standard usage, types of and compliance with standards

Features of dynamic inter-organizational business processes:
[28] introduced the concept of "dynamic inter-organizational business processes". It entails a continuous building and dismantling of partnerships, which is the key difference between dynamic and static inter-organizational business processes, which have existed since the 1960s and have been based on technologies such as EDI. This study introduces six requirements of dynamic inter-organizational business processes as bellow:

1. Process Autonomy: capability of autonomously defining and executing business processes
2. Flexibility: dynamic inter-organizational business processes are subject to more changes than static inter-organizational business processes
3. Scalability: provides quick support for adding business processes or for increasing significantly the instances of an already existing business process
4. Service Autarchy: capability of independently associating the services needed to support business processes or sub-processes
5. Service Extensibility and Integration: new services must be continually integrated due to changes in the business processes
6. Asynchronous Service Evolution: with dynamic inter-organizational business processes, services evolve independently

In synthesis and according to the above, we can conclude that the crucial requirements needed to successfully implement an IOISs are:
1) From the IT point of view:

- *Flexibility and scalability*, i.e. the capability of plugging-in and unplugging software/business components based on market needs as well as
- *Interoperability and integration*, creating a unified whole by interconnecting different systems or components.

2) From the business point of view: *process-centric architecture and continuous business process improvement,* i.e. the capability of executing all type of business processes based on a universal standard.

It appears that these requirements are the same of those of business agility, a concept that extends adaptability and flexibility to include speed and scalability [2], and a

term applied to the capability of an organization to react to unanticipated market change and rapid transformation of business processes [10]. This translates into the fact that the strategic value of IOIS for partners mostly depends on the alignment between IOIS specifications and capabilities with the requirement of business agility.

3   Compatibility of SOA Features with IOIS Requirements

The service-oriented architecture (SOA) approach is the latest in a long series of attempts in software engineering in trying to foster the reuse of software components [19]. SOA is an enterprise architecture that advocates loosely coupled and reusable systems. It has evolved from component-based development to distributed internet architectures as a new abstraction layer allowing internal and external systems to interact using common standards and protocols [4].

In a SOA, the business and technical processes are implemented as services. Each service represents a certain functionality that maps explicitly to a step in a business process. In this context, a service is a software component that can be reused by another software component or accessed via a standard-based interface over the network [22].

Technically, the main components of SOA are: service directory, service provider and service requester (see Figure 1). Some of the most important standards related to these components are:

- SOAP (Simple Object Access Protocol) (www.w3.org/TR/SOAP12): defines a mechanism for the communication with Web services over the Internet. It specifies the format of messages that are exchanged between the service requestor, the service provider, and the service directory.
- WSDL (Web Services Description Language) (www.w3.org/TR/WSDL.html): defines the structure and the contents of the service directory. Two related pieces of this standard are: 1) service types which are typically standardized by standards and 2) actual business information.
- UDDI (Universal Description, Discovery, and Integration) (www.uddi.org): provides the capability to describe a Web service, without the need to have it formally standardized. A WSDL description of a Web service provides all information needed to actually invoke it [19].

The fundamental features of SOA are: (1 it is based on services that can be readily integrated, (2 it is based on standards, (3 it is available on multiple platforms, (4 it provides self-contained (hence, loosely coupled) services, and (5 it incorporates and presupposes a contract that specifies the functionalities offered and at the same time, guarantees that they are replicable [1, 23].

A close examine of SOA special features and characteristics shows that they are fully compatible with the special requirements of IOISs listed above.

Furthermore, [22] illustrates the SOA potential benefits to IOISs as follows: loosely coupled applications and location transparency, application connectivity and interoperability, alignment of IT around the needs of the business, enhanced reuse of existing assets and applications, process-centric architecture, parallel and independent development, better scalability and graceful evolutionary changes, reduced costs of application development and integration, easier maintenance and reduced vendor

lock-ins. In turn, [9] claims that from a technical point of view, SOA provides flexible and cost-effective solutions through interoperability, reusability and composability between loosely coupled services, while from a business point of view, SOA is an architectural strategy that helps achieve enterprise resource integration and organizational agility.
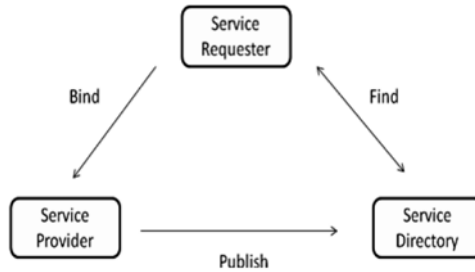

Fig. 1. Components of SOA

The SOA has been chosen as it allows for increased flexibility and scalability in comparison to traditional software engineering techniques [11].

SOA, web services and Web-based technology platforms provide best hope at IT level and provide reasonable solutions for reaching technical interoperability [8] allowing different distributed services to run on a variety of software platforms and hardware architectures. In effect, service composition enables scalability from small to more complex, distributed applications. Also, the SOA has the potential to address application integration issues by opening up the functionality with standardized, interoperable interfaces. Since services may be offered by different participants and communicated via the Web, they provide a distributed computing infrastructure for both intra- and inter-enterprise application integration and collaboration [26]. In terms of integration with legacy systems, SOA facilitates aligning existing IT infrastructure and systems to achieve end-to-end enterprise connectivity by removing redundancies, generating unified collaboration tools, and streamlining IT processes [3].

Last but not least, for process-centric architecture and continuous business process improvement, SOA allows a clear representation of the process flows identified by the order of the components used in a particular business service—and provides business users with an ideal environment for monitoring business operations. The process is decomposed into a series of steps, each representing a business service. In effect, each service or component functions as a sub-application. These sub-applications are chained together to create a process flow capable of satisfying the business need. This granularity lets processes leverage and reuse each sub-application throughout the organization [6].

Accordingly, SOA contributes flexibility and scalability [9, 11, 21, 24], interoperability and integration [8, 9, 21, 24] and process-centric architecture [6, 21] to the architecture of the IOISs in order to cover all the requirements of this kind of information systems. This demonstrates that SOA is today be one of the best architecture for developing IOISs, leading us to propose a SO-based architecture for IOISs that should be capable for developing a seamless IOIS for supporting agile businesses needs. The

architecture of service-oriented IOISs, consequently takes advantage of all the bene-fits and advantages of the SOA - as the latest approach in software engineering trend-. In comparison with SOA, Model-Driven Architecture (MDA) brings another perspec-tive which is not an alternative approach to SOA because these two approaches enable and complete each other [17, 25, 31].

4 Service Oriented-Based Architecture for IOIS

Based on the excellent fit between SOA features and IOISs requirements, we propose a three layer architecture for IOISs. Ours is a top-down architecture span from busi-ness to IT infrastructure in order to support business requirements. As the two others are designed based on it, the business service layer is the most important layer of the proposed architecture. Composite applications, constituting the second layer, need to be identified for supporting business services throughout the organization as well as services between organizations. As the lowest layer, integration between applications by utilizing service-oriented standards will be investigated (see Figure 2).

The Business service Layer. The business architecture is the vital part and the first architectural layer of an IOIS that impacts other architectural layers. In this layer we have to decompose business processes in order to determine business services. When applying the service-oriented paradigm to enterprise computing, the functionality is encapsulated as a service, either as a simple or composite one [27].
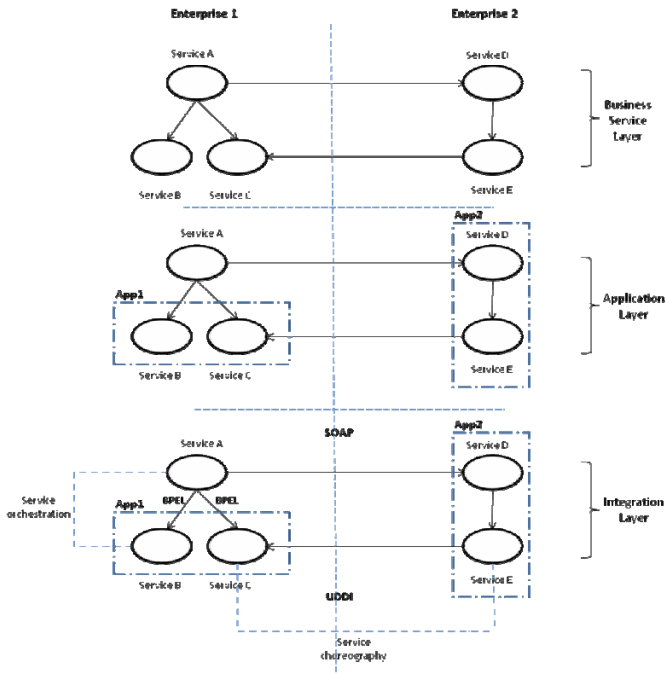

Fig. 2. The proposed Service-Oriented IOIS Architecture

A business process is composed of a series of continuous actions or operations that are performed upon a commodity. It is usually initiated by a customer. It must provide results directly to a customer, who may be internal or external to the company [18]. We have observed three common patterns for studying business process: 1) *Organizational level* consists of strategic level, tactical level and operational level [18], 2) *Interaction* consists of peer-to-peer/ conversational structure and hierarchical structure [14, 19], 3) *Value chain* consists of primary and support business processes [24]. According to the requirements of agile businesses, that also describe the requirements of developing an IOIS, the componentization of the business using the SOA is not far.

In this regard, we should identify the functionality of each business process, the so-called aspect. Every business process can feature several aspects: 1) the functional aspect which describes how a business process is composed of sub-processes 2) the control aspect which describes how activities are executed dependent on the result or completion of other activities 3) the organizational aspect in which the relation between the business process and the organization structure is established and 4) the operational aspect which describes external services to be used during the process. These aspects are the atomic part of business processes and contain only one functionality and shape the representation element of each business process [28]. Therefore, for following service-oriented logic in IOIS architecture, we need to use a tool or standard for converting business processes into business services.

BPEL4WS is a technology that defines a business process as a set of service invocations [6]. It is a service orchestration and choreography language which fulfills the need for modeling all the above mentioned business process patterns. By using this language, we can define application and infrastructure services directly based upon business services.

Application Layer. We propose a composite approach for defining applications. In other word, we have combined different services for determining application services. A composite application is created by a set of interconnected and specialized services. The specialization information contains both parameterization and connection information. Parameterization information adapts the service to the individual needs of the composite application. The connection information contains information about the connection of the service with other services within the context of the composite application [28].

In effect, composite applications are similar to Component Based Software (CBS) focusing on building large software systems by integrating previously build software components. A composite application has three different layers:

- User interface layer: the different forms and functions are presented to the user
- Choreography layer: defines the correct order of calling services in the composite application
- Service layer: represents service to composite application [20].

We use BPEL4WS because it fully supports this compositional approach. BPEL4WS's composition functionality described by [14] as following: enable flexible integration, offering choreography as a web service, support for multiple composition patterns and support for lifecycle management. It is currently the preferred standard for performing web service composition [27].

Integration Layer. After determining composite applications, in the integration layer we will identify the environment in which application services supports all the business services.

The relationship between services in composite applications could be inter or intra-organizational. Hence the information about existing services is stored in a service registry based on UDDI. For each representation element of the business processes, an appropriate service has to be found. In most cases, there is no perfect fit; therefore, the search has to include services which can be properly adapted. Furthermore one service may fit well with several representation elements. This information is returned in the form of WSDL files and by using XML-based interface. These XML-based messages do exchange between services by using SOAP. Service orchestration and choreography within organization and between organization and its stakeholders is handled by BPEL4WS [20, 28]. Apart from the application integration, some other integration considerations are the end-user interface, the application connectivity, the process integration, the information integration and a build-to-integrate development model [6].

We could classify different integration standards based on the layer of integration. Figure 3 illustrates the related standards of every integration layer [15].
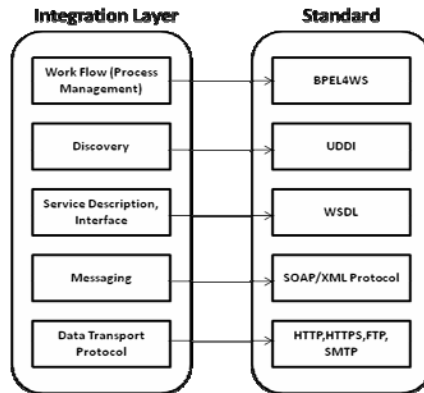

Fig. 3. Integration standards

5   Conclusion

Inter-Organizational Information Systems (IOISs), a cross-organizational automated information system, allows sharing an information system between several companies thus facilitating the creation, storage, transformation and transmission of information. Based on the literature in this field, we have demonstrated that an IOIS is capable of 1) plugging-in and unplugging software/business components (flexibility and scalability), 2) interconnecting different systems or components (interoperability and integration) and 3) executing all types of business processes based on a universal standard (process-centric architecture and continuous business process improvement).

The strategic effects of IT on IOISs on one hand, and the introduction of the service-oriented architecture in software engineering on the other hand, led us to study the compatibility of SOA's features and capabilities with the requirements of an IOIS. In SOA, the business and technical processes are implemented as services; a service being a software component that can be reused by another software component or accessed via a standard-based interface over the network. We determined that SOA provides flexible and cost-effective solutions through interoperability, reusability and composability between loosely coupled services. We discussed the fact that it also helps achieve enterprise resource integration and organizational agility based on interoperability and addresses application integration issues by opening up the functionality with standardized, interoperable interfaces. We finally showed that SOA is a process-centric architecture because it decomposes business processes into business services, each business service functioning as a sub-application.

After making sure there is a real compatibility between the IOISs requirements and the SOA's features, we have proposed a top-down service-oriented architecture for IOISs composed of a three layers architecture: 1) business service layer that consists of decomposing business processes in order to determine business services, 2) application layer that consists of combining different services for determining application services 3) integration layer that consists of identifying the environment in which application services supports all the business services. In order to fully support the compositional approach for defining and integrating application services of our three layers model, we recommend BPEL4WS as an orchestration and choreography language in the business service layer.

As a next step, we will examine the combination of the artifacts of this architecture with the building blocks of Enterprise Architecture (EA) in order to follow this approach from strategic level of the company and inter-organizational business processes with business partners. We will also investigate this approach from the Business Process Management (BPM) perspective for supporting inter-organizational BPM systems and work-flow management systems.

References

1. Baskerville, B., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M., Virili, F.: Extensible Architectures: The Strategic Value of Service Oriented Architecture in Banking. In: Proceeding of European Conference on Information Systems, ECIS (2005)
2. Baskerville, R.: Business Agility and Information Technology Diffusion. IFIP Advances in Information and Communication Technology, vol. 180, Atlanta, Georgia (2005)
3. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of Service-Oriented Architecture on Enterprise Systems, Organizational Structure and Individuals. IBM Systems Journal 44(4) (2005)
4. Brahe, S.: BPM on top of SOA: Experiences from the Financial Industry. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 96–111. Springer, Heidelberg (2007)
5. Cash, J.I., Knsynski, B.R.: IS Redraws Competitive Boundaries. Harvard Business Review, 134–142 (March-April 1985)
6. Channabasavaiah, K., Holley, K.: Migration to a Service-Oriented Architecture. IBM Software Group (2004)

7. Chatterjee, D., Ravichandran, T.: Inter-organizational Information Systems Research: A Critical Review and an Integrative Framework. In: Proceedings of the 37th Hawaii International Conference on System Sciences, USA (2004)
8. Chen, D., Doumeings, G., Vernadat, F.: Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry 59, 647–659 (2008)
9. Chen, H.M.: Towards Service Engineering: Service Orientation and Business-IT Alignment. In: Proceedings of the 41st Hawaii Int. Conference on System Sciences (2008)
10. El-Ghareeb, H.A.M.: Aligning Service Oriented Architecture and Business Process Management Systems to Achieve Business Agility. Requirements Network Group (RQNG) (January 2009)
11. Hanemann, A., et al.: PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 241–254. Springer, Heidelberg (2005)
12. Johnston, H.R., Vitale, M.R.: Creating Competitive Advantage with Interorganizational Information Systems. MIS Quarterly (June 1988)
13. Kaufman, F.: Data systems that cross company boundaries. Harvard Business Review (January-February 1966)
14. Khalaf, R., Mukhi, N., Weerawarana, S.: Service-Oriented Composition in BPEL4WS. In: Proceeding of the twelfth international World Wide Web conference, WWW 2003, Budapest, Hungry (May 2003)
15. Koshutanski, H., Massacci, F.: An Access Control Framework for Business Processes for Web Services. In: Proc. 2003 ACM workshop on XML security, Virginia, USA, pp. 15–24 (2003)
16. Kumar, K., Dissel, H.G.V.: Sustainable Collaboration: Managing Conflict and Cooperation in Interorganizational Systems. MIS Quarterly (September 1996)
17. Kumaran, S., Bishop, P., Chao, T., Dhoolia, P., Jain, P., Jaluka, R., Ludwig, H., Moyer, A., Nigam, A.: Using a Model Driven Transformational Approach and Service-Oriented Architecture for Serving Delivery Management. IBM Systems J. 46(3), 513–529 (2007)
18. Kung, M.T., Zhang, Y.Z.: Analysis of Business Process Models in Enterprise Web Services. International Journal of E-Business Research 4(2), 69–87 (2008)
19. Leymann, F., Roller, D., Schmidt, M.T.: Web services and business process management. IBM Systems Journal 41(2) (2002)
20. Liegl, P.: The Strategic Impact of Service Oriented Architecture. In: Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS 2007), Tucson, AZ (2007)
21. Lu, X.H., Huang, L.H., Heng, M.S.H.: Critical success factors of inter-organizational information systems—A case study of Cisco and Xiao Tong in China. Information & Management 43, 395–408 (2006)
22. Mahmood, Z.: Service Oriented Architecture: Potential Benefits and Challenges. In: Proceedings of the 11th WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, July 26-28 (2007)
23. Meredith, L.G., Bjorg, S.: Contracts and Types. Com. of the ACM 46(10), 41–47 (2003)
24. Porter, M.: Changing Patterns of International Competition. California Management Review XXVIII(2), 9–40 (Winter 1986)
25. Radhakrishnan, R., Wookey, M.: Model Driven Architecture Enabling Service Oriented Architecture. Sun Micro Systems (March 2004)
26. Raghupathi, W., Kesh, S.: Interoperable Electronic Health Records Design: Towards a Service-Oriented Architecture. e-Service Journal 5(3), 39–57 (2007)

27. Rosenberg, F., Dustdar, S.: Business Rules Integration in BPEL:A Service-Oriented Approach. In: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, Washington, DC, USA, pp. 476–479 (2005)
28. Schmidt, R.: Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes. In: Jeckle, M. (LJ) Zhang, L.-J. (eds.) ICWS-Europe 2003. LNCS, vol. 2853, pp. 123–136. Springer, Heidelberg (2003)
29. Sprague, R.H., McNulin, B.C.: Information System Management in Practice. Prentice Hall, Englewood Cliffs (1993)
30. Stelzer, D., Fischer, D., Nirsberger, I.: A Framework for Assessing Inter-Organizational Integration of Business Information Systems. Int J. Interoperability in Business Information Systems 2(2), 9–20 (2006)
31. Torkaman Rahmani, A., Rafe, V., Sedighian, S., Abbaspour, A.: An MDA-Based Modeling and Design of Service Oriented Architecture. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 578–585. Springer, Heidelberg (2006)