



Web Service Oriented Architecture Modeling With Pattern for Electronic Business Organization

Jitendra Joshi #1* Nisha Singh#2 Manjari kumara #3

Department of Computer Science and Engineering,
Jayoti Vidyapeeth Women's University,
Vedaant Gyan Valley, Village Jharna, Mahala - Jobner, Link Road,
Jaipur Ajmer Express Way, NH-8, Jaipur, Rajasthan
(India) -303007

Abstract— *In this paper we initiate how the Patterns for electronic business admiring comment the service-oriented architecture approach. We look at how service-oriented architecture approaches apply to Patterns, in particular the Self-Service and comprehensive enterprise business patterns, the application integration pattern, and the associated application patterns, runtime patterns, and Product mappings.*

Keywords— *Web service, oriented architecture, modeling, electronic business, extended enterprise, EA I.*

I. INTRODUCTION

A web service-oriented architecture (SOA) is a set of principles or philosophy and methodologies for designing and developing model in the form of interoperable services. These services are well-defined business functionalities that are built as model components that can be reused for different purposes. Service-oriented architecture modeling principles are used during the phases of model development and integration.

Service-oriented architecture defines how to integrate widely disparate applications for a Web-based environment in electronic business and uses multiple implementation platforms in. Rather than defining an application integration pattern for extended enterprise, Service-oriented architecture defines the interface in terms of protocols for electronic business and functionality.

Patterns for electronic business can be used with any method. When used with a service oriented architecture approach, the focus is on create and integrating insecurely joined services as a substitute of applications to maintain a business and industries. The extended enterprise business pattern addresses communications and collaborations between business processes in separate enterprises.

II. Using services oriented architecture model with pattern for electronic business organization

The Patterns for electronic-business is not a solution development methodology. It is a collection of proven architectures model, derivative from more than 25,000 successful Internet-based arrangements. The Patterns for electronic business connection the business and information technology gap by significant architectural patterns at the different levels from business patterns to application patterns to runtime patterns.

With a service-oriented architecture approach, this is taken to another level of abstraction with the creation of a service integration layer. This enables a business to be agile and respond quickly and efficiently to changes in the market and its customer's requirements, as well as to stay competitive.

Patterns for electronic business can be used with any method. When used with a service oriented architecture approach, the focus is on create and integrating insecurely joined services as a substitute of applications to maintain a business and industries. The service-oriented paradigm leverages the opinion of services as discrete structure blocks of business and industries functionality which are composed to satisfy business and industries requirements. Services are self-contained and modular, while applications are likely to be also coarse-grained to be reused successfully, and are regularly as well nonflexible to be leveraged inside an enterprise or a comprehensive enterprise.

The on hand business, integration, application and runtime patterns are consistent with the service oriented architecture approach. The business difficulty will force the identification of the appropriate business, integration, application and runtime patterns concerned in an impending solution. Runtime patterns that involve two or more middleware nodes connecting logical application tiers will have additional communication options between the tiers with the use of services in service oriented architecture.

One candidate for a new compound pattern in the framework of services is the value-chain. Basically, the value-chain consists of a self-service business and industries pattern and a set of recurring extended enterprise business patterns that can be combined to provide value to a chain of business partners. This end-to-end composite pattern is prevalent in any value-net.

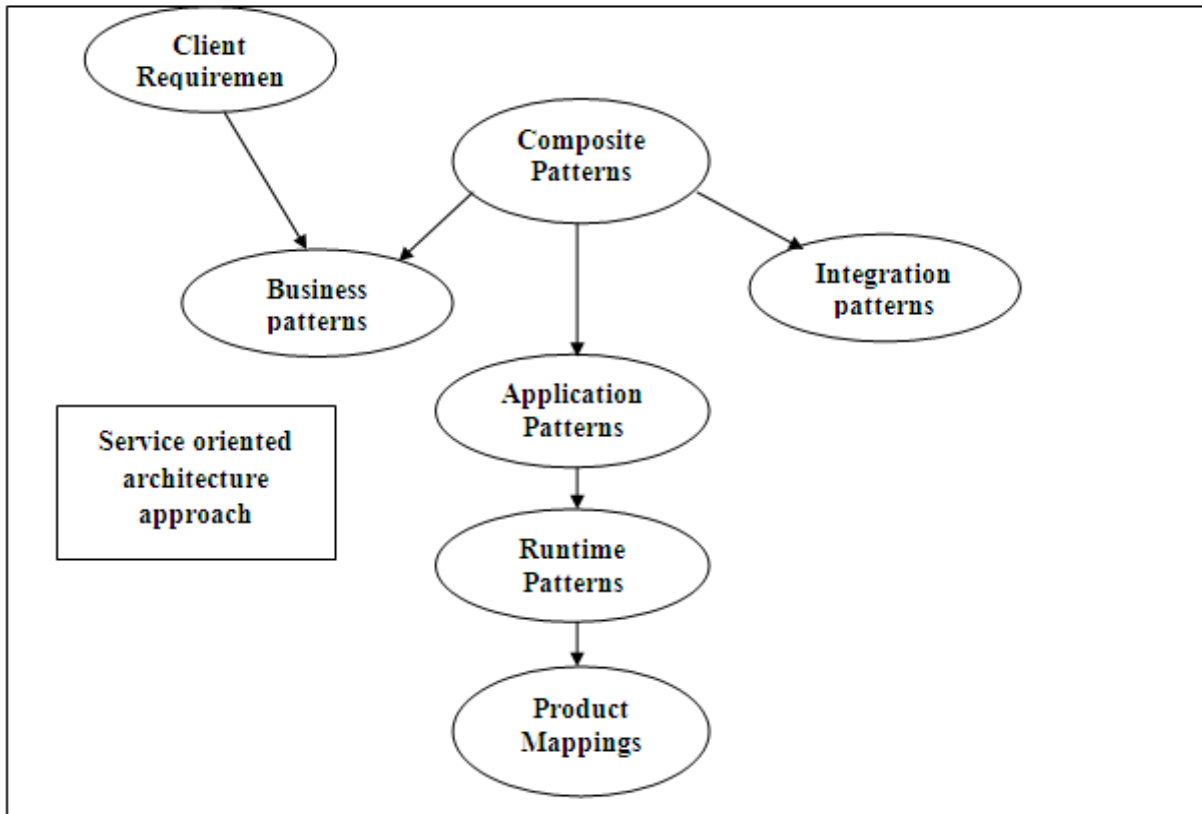


Figure 2.1.: Services Oriented Architecture Model with Pattern for electronic business organization

III. Self services business pattern model

The Self-Service business pattern, we be capable of say the User-to-Business or U2B model, captures the real meaning of direct relations between concerned parties and a business. Interested parties include clients, business partners, industry partners, stakeholders, staff, and all other persons with whom the business intends to interact. For simplicity, these interested parties are referred to as users. In this meaning business represents a variety of types of organizations together with big enterprises, small and medium businesses, government agencies.

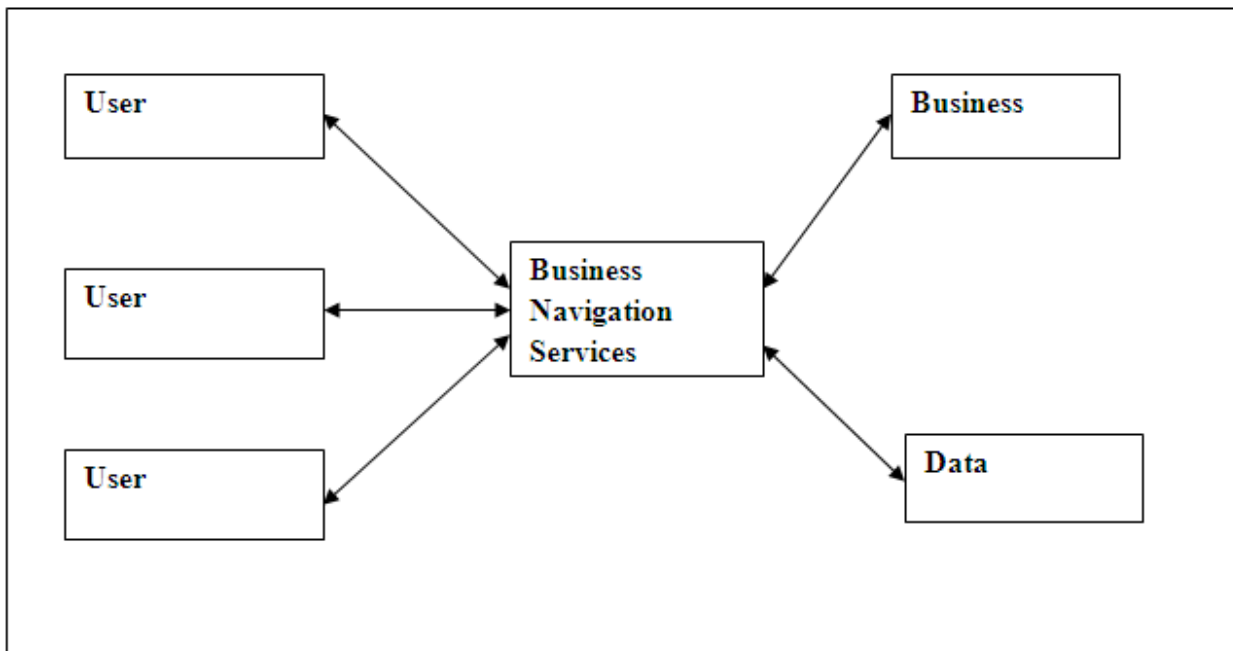


Figure 3.1: User-to-Business or U2B model

The Self-Service: Openly incorporated single channel application pattern, for example, provides a user access channel to presentation logic running in the presentation layer. The presentation layer can call for or consume services provided on the Web application layer. The Web application layer in turn can consume services provided on the back-end or enterprise layer, which is show below figure.

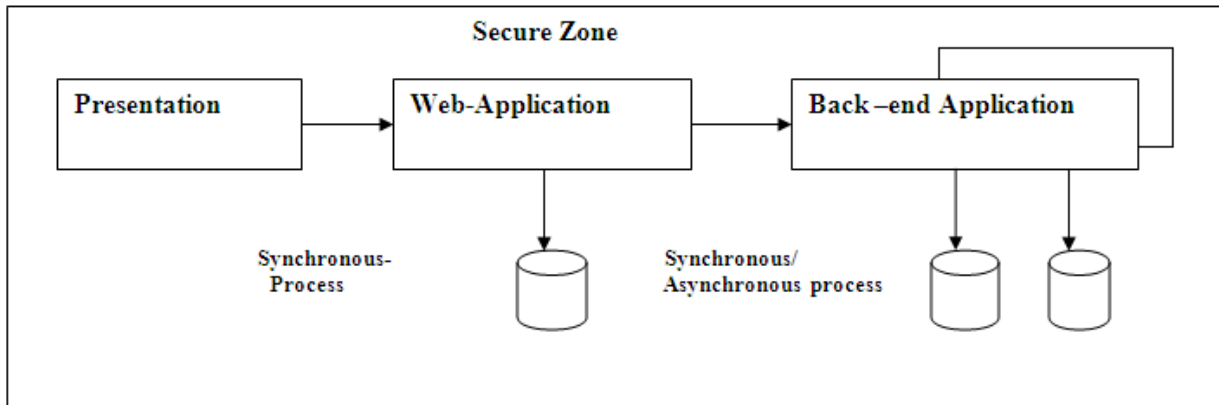


Figure 3.2: Self service model with Directly Integrated Single Channel application pattern

IV. Extended enterprise business pattern

The Extended Enterprise business pattern addresses communications and collaborations between business processes in separate enterprises. The interactions and collaborations are implemented using programmatic interfaces to join inter-enterprise applications or services. In this case, an enterprise can be both a service consumer and a service provider.

The Extended Enterprise: open to the elements through relationship application pattern, for example, allows a pair of applications to in a straight line converse with each other crosswise organization limitations. A resource application in Partner A can request or use services provided by Partner B, as shown in below Figure even though not exposed on this illustration, a resource application in Partner B can also call for or use services provided by Partner A.

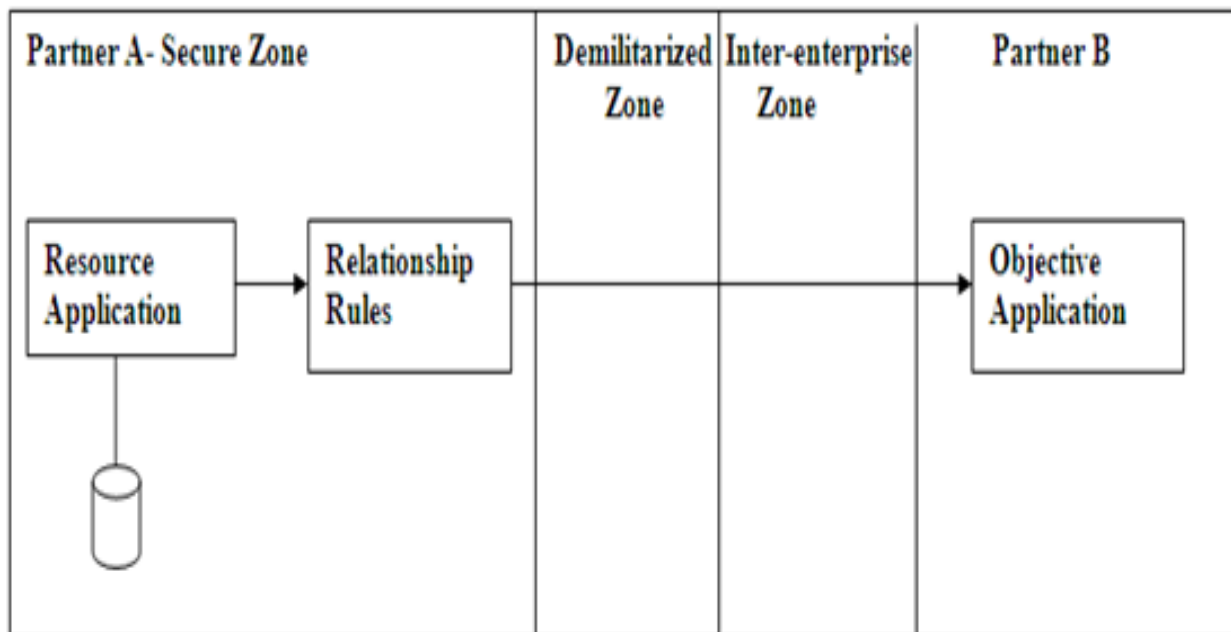


Figure 4.1: Extended enterprise to exposed direct connection application pattern

Extended Enterprise is basically a demonstration of the Application Integration pattern with supplementary QoS concerns such as security, performance and accessibility. A service-oriented structural design implemented with open standards facilitates this extension by simplifying the attempt to integrate likely different technologies and information models.

V. Application integration pattern for Extended Enterprise

The Application Integration patterns imprison normally experimental resolution alternatives in the sphere of influence of Enterprise Application Integration (EAI). They confine best practices in the region of back-end integration of applications and data, process automation, and workflow implementations connecting human interactions. It is

imperative to note that front-end integration such as the composition of a portal or single sign-on across multiple applications is captured by the access integration pattern.

Application patterns for Extended Enterprise are basically the implementation of Application Integration across organizational boundaries. The differentiation is mainly in the way Quality of Service aspects affect the Runtime patterns.

S. No.	Facilities/Service	Example
1.	To Buy Side	Direct Procurement Indirect Procurement Provide Series Finishing
2.	To Sell Side	Business to Business Electronic Commerce
3.	To Trading collaborator renovation	Electronic data interchange transformation
4.	To Exchange Participation	Personal Electronic-Exchanges Community Electronic-Exchanges

Table: 1. you can notice several cross-industry examples of the Extended Enterprise pattern.

We provide list some industry-specific example applications that can be implemented though the Extended Enterprise pattern.

S. No.	Industry	Example
1.	Industrialized	Provide Series preparation provide series finishing Vendor-Managed Inventory
2.	Journey	Inspection flight or area accessibility Building or modifying reservations
3.	put up for sale as retail	Inspection provider catalog Placing replacement orders Paying suppliers automatically
4.	Financial/ monetary	transfer payments/cost inspection account balances Obtaining credit information Finance Origination or committee dealing out securities
5.	Telecommunication	OSS Integration Cross organization order management Managed service provider interconnect

Table: 2 Industry-specific examples

VI. Conclusion

To improve the reuse of Web Service oriented architecture modeling at the business or enterprise level, architects must define a strategy for publishing and providing facilities to access services information for electronic business. These services are well-defined business functionalities that are built as model components that can be reused for different purposes.

The patterns for electronic business can be used with a service oriented architecture approach, the focus is on create and integrating insecurely joined services as a substitute of applications to maintain a business and industries. In this paper, we have also discussed Application integration pattern for extended enterprise and we have provide idea for the extended enterprise it's to open to the elements through relationship application pattern for business or organization.

References

- [1] Ivar Jacobson, 1992, Object-oriented Software Engineering - a Use Case Driven Approach, Addison Wesley, ISBN 0201544350.
- [2] S. Bradner, March 1997, Key words for use in RFCs to Indicate Requirement Levels, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119.
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, 1995, Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, ISBN 0-201-63361-2.
- [4] Paul Allen, 1998, Component-based Development for Enterprise Systems, Cambridge University Press, ISBN 0521649994.
- [5] Jonathan Adams, Srinivas Koushik, Guru Vasudeva, George Galambos, 2001, Patterns for e-business: A Strategy for Reuse, IBM Press, ISBN 1-931182-02-7.
- [6] Ali Arsanjani, 2001. A Domain-language Approach to Designing Dynamic Enterprise Component based Architectures to Support Business Services, Proceedings of Technology of Object-oriented Languages and Systems 39.
- [7] Craig Larman, 2001, Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design, 2nd Ed, Prentice Hall, ISBN 0130925691.

- [8] Parrish, Allen, Brandon Dixon, and David Cordes, 2001, A Conceptual Foundation for Component-Based Software Deployment, *Journal of Systems and Software*. Vol. 57, Issue 3, 193–200.
- [9] Keith Levi, Ali Arsanjani, 2002, A goal-driven approach to enterprise component identification and specification, *Communications of the ACM* Volume 45 Issue 10.
- [10] Lawrence Wilkes, 2002. *Business Integration - Drivers and Directions*, CBDI Forum.
- [11] Olaf Zimmermann, Mark R Tomlinson, Stefan Peuser, 2003 *Perspectives on Web Services; Applying SOAP, WSDL and UDDI to Real-World Projects*, Springer, ISBN 3-540-00914-0.
- [12] van der Aalst, W. M. P., and A. J. M. M Weijters. 2004, *Process Mining: A Research Agenda*, *Computers in Industry*. Vol. 53, Issue 3 (doi: 10.1016/j.compind.2003.10.001), PP. 231–244.
- [13] Dustdar, Schahram, and Wolfgang Schreiner, 2005, *A Survey on Web Services Composition*, *International Journal of Web and Grid Services (IJWGS)*. Vol. 1, Issue 1, PP. 1–30.
- [14] Jøsang, Audun, Roslan Ismail, and Colin Boyd, 2007, *A Survey of Trust and Reputation Systems for Online Service Provision*, *Decision Support Systems*. Vol. 43, Issue 2, Pp. 618–644.
- [15] Li, Xitong, Yushun Fan, Jian Wang, Li Wang, and Feng Jiang, 2008, *A Pattern-Based Approach to Development of Service Mediators for Protocol Mediation*, *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, PP. 137–146.
- [16] Juric, Matjaz B., Ana Sasa, Bostjan Brumen, and Ivan Rozman, 2009, *WSDL and UDDI Extensions for Version Support in Web Services*, *Journal of Systems and Software*. Vol. 82, Issue 8 (doi:10.1016/j.jss.2009.03.001), PP. 1326–1343.
- [17] Clark, Allan, Stephen Gilmore, and Mirco Tribastone, 2009, *Quantitative Analysis of Web Services Using SRMC, SFM*. Vol. 5569, PP. 296–339.