



# Service oriented Architecture for Business Dynamics: An Agent-based Approach

O. P. Rishi<sup>1\*</sup>, Ashok Sharma<sup>1</sup>, Archana Bhatnagar<sup>1</sup> and Ashutosh Gupta<sup>2</sup>

## ABSTRACT

*In today's rapidly changing environment the industries are interested in executing business functions that has scope in multiple applications. Business dynamics and technological innovations have felt organizations to comply with a disparate mix of operating systems, applications and databases. This makes it difficult, time-consuming and costly for IT departments to deliver new applications that integrate heterogeneous technologies. It demands high inter-operability and more flexible and adaptive E governance process management. The inclination is to have systems assembled, from a loosely coupled collection of Web services, which are universal and integrated. This technical area appears to have scope where the Agent Technology can be exploited with significant advantages. With Service Oriented Architecture a decomposable architecture, and associated set of development and IT management disciplines, composed of loosely coupled services communicating via pre-established protocols, these services can be assembled ad-hoc to form customized applications that address a wide variety of business requirements. A conceptual framework is being proposed in the course of agent-based Service Oriented Architecture (SOA) for e-governance. In which we try to integrate Service Oriented Architecture with the agent technology & other tactical technologies like web services, e-governance workflow services, e-governance meta-rules, search optimization of services and semantic Web technology for business service mappings.*

**Keywords:** Multi-agent systems; Service oriented architecture; e-governance workflow & services; e-governance dynamics.

## 1. Introduction

Today the technology world believes that adoption of a e-governance Oriented Architecture (SOA) paradigm is strategic and should be part of the most software projects. Agent technology is considered to be the most successful technology supporting Service Oriented Architecture. It is known that an Agent technology is used to implement complex systems and applications that are communication-centric, based on distributed computational and information systems, and requiring autonomous components readily adaptable to changes. Agent plays the role of efficiently supporting distributed computing and allows the dynamically composition of Web services. Now it is desired that agent technology integrate with other enterprise computing technologies to improve the computational proficiency. There are several unseen technical issues and the existing technology has significant limitations. Yet, the prototype systems based on

---

<sup>1</sup> Birla Institute of Technology, Mesra , Ranchi, BIT Jaipur Campus, Jaipur, Rajasthan 302017, India

<sup>2</sup> Sir Padampat Singhanian University, Udaipur, Rajasthan, India

\* Corresponding Author: (E-mail : omprakashrishi@yahoo.com)

the underlying infrastructure can help to increase awareness of these issues and to set down possible solutions. In an agent based Service Oriented Architecture approach the scenario would be characterized mainly by three actors: Service Providers, e- governance process Manager and Users as shown in figure seven.

## **2. Introduction to Service Oriented Architecture and SOA for e-governance**

Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different domains. SOA is an approach or strategy in which applications rely on services available in a network such as the World Wide Web. And it can be considered as a way of sharing functions (typically business functions) in a widespread and flexible way. In other words we can say, SOA is a service-oriented architecture and can be defined as a group of services, which communicate with each other. It uses services available in a network and promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services; where service is an implementation of business functionality, and such service can be utilized by users in different applications or e-governance processes.

In SOA framework, Service modeling includes:

- Service Oriented Enterprise
- Service Oriented Architecture
- Service Oriented Computing

**SO Enterprise:** The Service Oriented Enterprise (SOE) is a new model for architecting software and I.T. infrastructure. It allows a business to view itself from the perspective of its customers, suppliers and other trading partners. The business value derived from this approach includes cost savings, flexibility and the ability to respond more quickly to marketplace changes.

**SO Architecture:** A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.

**SO Computing:** Service-oriented computing provides a way to create a new architecture that reflects components' tendencies toward autonomy and heterogeneity.

Normally the e-governance process environment, which does not use the service, oriented architecture lack the interaction of multiple services at the same time to exchange messages or to perform some task. Using the SOA environment the following benefits can be drawn:

- Reuse of services enabled by the decoupling of service providers and service consumers
- Structured description of interfaces
- Discoverability of services through the registry
- Incremental deployment and maintenance
- Architectural partitioning that allows the service provider to be modified or even replaced without impact to the service consumer
- Flexibility and agility is facilitated by allowing multiple services to be composed quickly into more complex services and allowing the process flow between services to be configured dynamically

Service-Oriented Architecture is preferred when there is a need for request-reply, real time integration between systems, and more than two systems are involved in the integration. Similarly Service-Oriented

Architecture is also preferred when a service being provided is a likely candidate for reuse, & service implementation requires no advanced knowledge of the service client.

Many challenges are faced when we adopt SOA. Managing services metadata, which includes exchange of messages to perform tasks, generating millions of messages, managing and providing information on how services interact is a complicated task. Lack of testing in SOA space, as today sophisticated tools are not available that provide testability of all headless services (including message and database services along with web services), no testing framework is available that would provide the visibility required to find the fault in the architecture and no provision for appropriate levels of security. The need of the proposed model / framework is arisen from the above challenges. Usually the design framework of SOA does not maintain or use agents, but in our proposed model we have tried to incorporate the service-oriented architecture (SOA) based on various business processes agent. Thereby making a model, which comprises of agent based service oriented architecture.

In Agent based SOA framework for e-governance; the following architectural principles for design and service definition focus on specific themes that influence the innate, behavior of a system:

- Many web-services are encapsulated to be used under the SOA Architecture.
- Services maintain a relationship that minimizes dependencies hence exhibiting the behavior of loose coupling.
- Services adhere to a communications agreement as defined in service contract.
- Logic of service is hidden / abstracted from the outside world.
- Logic is divided into services with the intention of promoting reuse.
- Collections of services can be coordinated and assembled to form composite services.
- Services have control over the logic encapsulated thereby exhibiting service autonomy.
- High-quality services are preferred than low-quality ones for service optimization.

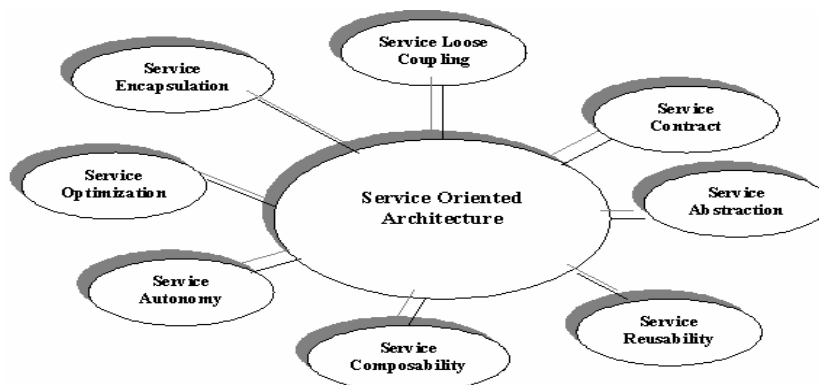
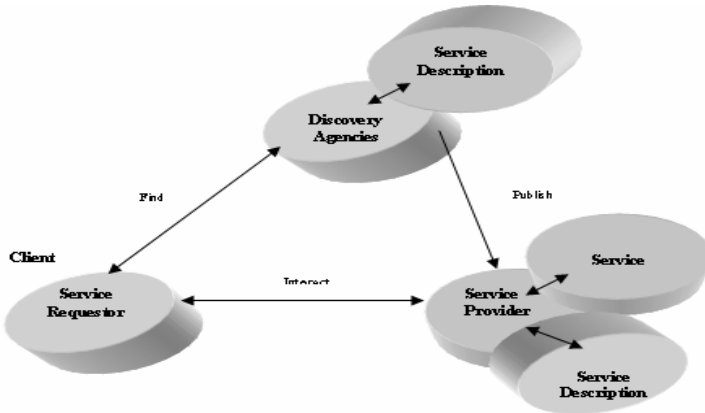


Figure 1: Service Components of SOA in Business Process

The functionality of SOA rotates around business processes and packaged as interoperable *services*. SOA also describes IT infrastructure which allows different applications to exchange data with one another as they participate in e-governance processes. The aim is to have loose coupling of services with operating systems, programming languages and other technologies. Web Services are the set of protocols by which Services can be published, discovered and used in a technology neutral, standard form. *Services* are what you connect together using Web Services. A service is the endpoint of a connection. Also, a service has some type of underlying computer system that supports the connection offered. Service is the important concept.

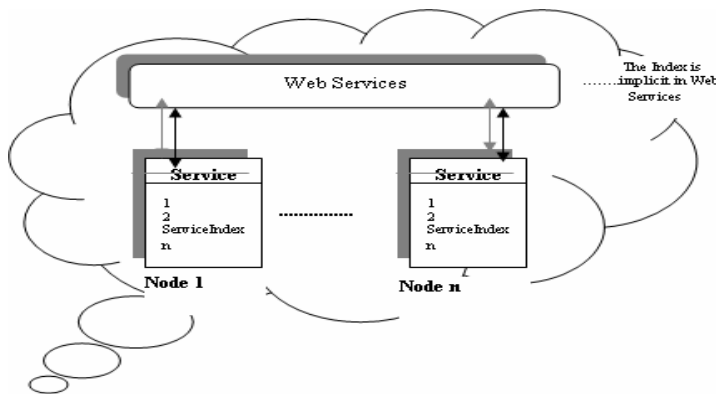


**Figure 2:** Services & Service Providers

SOA separates functions into distinct units, or services, and makes them available on a network so that they can be combined and reused in the business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services. SOA concepts usually built upon older concepts of distributed computing and modular programming.

### 3. Service Oriented Architecture and Web Technology

The technology of Web services is connection technology for service-oriented architectures. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in a way that is understandable to both the service consumer and service provider. A service provider can also be a service e-governance user. The term Web Services refers to the technologies that allow for making connections. Services are what we connect together using Web Services. A service is the endpoint of a connection and has some type of underlying computer system that supports the connection offered. The combination of services - internal and external to an organization - makes a service-oriented architecture.



**Figure 3:** e-Governance Services and Web Technology in SOA framework

In general, business entities offer capabilities and act as service providers. One who makes use of services

is referred to as service consumers. The service description allows prospective consumers to decide if the service is suitable for their current needs. Although SOA is commonly implemented using Web services, services can be made visible, support interaction, and generate effects through other implementation strategies. Web service-based architectures and technologies are specific and concrete.

#### 4. Web Services and Agent Based Service Oriented Architecture

In web service-based architectures the service providers can register the instance of the service in the registry making it available to service consumers. The service consumer may then query the registry in order to retrieve the binding information required to access the service. The service consumer then invokes the service. The relationship between a service provider and consumer is dynamic and established at runtime by a binding mechanism. Dynamic binding minimizes the dependencies between the service consumer and the service provider. Service Oriented Environment is based on the following major principals:

- Service is the important concept. Web Services are the set of protocols by which Services can be published, discovered and used in a technology neutral, standard form.
- SOA is not just architecture of services seen from a technology perspective, but the policies, practices, and frameworks by which we ensure the *right* services are provided and utilized.
- With SOA it is critical to implement processes that ensure that there are at least two different and separate processes—for provider and user.
- Rather than leaving developers to discover individual services and put them into context, the e-Governance Service Bus. It is instead their starting point that guides them to a coherent set that has been assembled for their domain.

The value of SOA is derived from the runtime and design/development/configuration activities. The development process gains speed by the reuse of services. Dynamic discovery and binding at runtime supports loose coupling leading to more stable and reliable applications

**Table 1:** Two sets of activities of the SOA pattern

<b>Design/Development/Configuration Events</b>	<b>Runtime Events</b>
<ul style="list-style-type: none"> <li>• it facilitates the reuse of the service and rapid development of service consumer applications</li> <li>• service interface is specified through the use of Web Service Definition Language (WSDL)</li> </ul>	<ul style="list-style-type: none"> <li>• client application accesses information about service which includes bindings supported by it and network address of server on which it is running</li> <li>• client has options to choose from variety of service providers which support the service interface, creating loose coupling between client application and specific service implementation being used</li> </ul>

Today, agents are being applied in a wide range of industrial applications. Most of the technology and market research companies, which provides their clients with advice about technology's impact on business and consumers, agree on the fact that the adoption of a SOA paradigm is strategic and should be part of the most forward-looking software projects. Agents who require a service from another agent enter into a negotiation for that service to obtain a mutually acceptable price, time, and degree of quality. Successful negotiations result in binding agreements between agents. This agent-based approach offers a number of advantages over more typical workflow solutions to this problem. The proactive nature of the agent means services can be scheduled in a just-in-time fashion (rather than pre-specified from the beginning), and the responsive nature of the agents means that service exceptions can be detected and handled in a flexible manner.

## **5. Proposed Model of Service Oriented Architecture for e-governance**

A service-oriented architecture (SOA) is an application topology in which the business logic of the application is organized in modules (services) with an identity, purpose and access interfaces. Services behave as "black boxes" where their internal design is independent of the nature and purpose of the requestor. In SOA, data and business logic are encapsulated in modular business components with documented interfaces. This helps to understand the design better and facilitates incremental development and future extensions. A SOA application can also be integrated with heterogeneous, external legacy and purchased applications more easily than a monolithic non-SOA application. Applications that have separate business layers are more suitable to access a SOA environment.

The proposed system is based on the emergent and more established technologies which we aim at integrating with agent technology, the need for SOA in organizations & agent-based SOA for business dynamics followed by the E governance process and the Behavior of system in SOA and the architecture related to web services. The proposed system consists of a number of specialized agents with different expertise. It comprises of the Web agent and Communication Service Agent (CSA), Application Interface Agent (AIA), Data Adaptation Agent (DAA) , Application agent and different E governance process and Data Retrieval agent and communication agent which are architected in order to work together for the optimized working using SOA.

The system architecture would be used in communities consisting of different kinds of agents like service providers, personal assistants and middle agents (e.g. service brokers, user profile managers, workflow managers, etc) and other agents like Communication Service Agent (CSA), Application Interface Agent (AIA), Data Adaptation Agent (DAA). These autonomous agents should be able to perform their tasks in cooperation or in competition with other agents and be able to interoperate with external entities (e.g., legacy software systems) for achieving their goals (semantic matching, service contracting etc.). They should have reasoning capabilities and support for dynamic behavior modification based on business rules. They should also be able to build workflows, compose the external Web services and monitor their execution. A distributed management should support the complete process.

The multiple agents that we have used have specific work and they work in co ordination with each other. When the user surfs on World Wide Web, he uses an agent to contact the Service Provider that in turn is using the Services associated with it in order to provide assistance. In our proposed model the Service Provider takes help of the E governance processManager whose task is to authenticate for services to Service Provider and define the Services for Business process.

We are defining the following agents to comprise services and connections between the services.

### **5.1 CSA (Communication Service Agent)**

Aim: to provide interface between Web server and the Website

Task (action) of the agent :

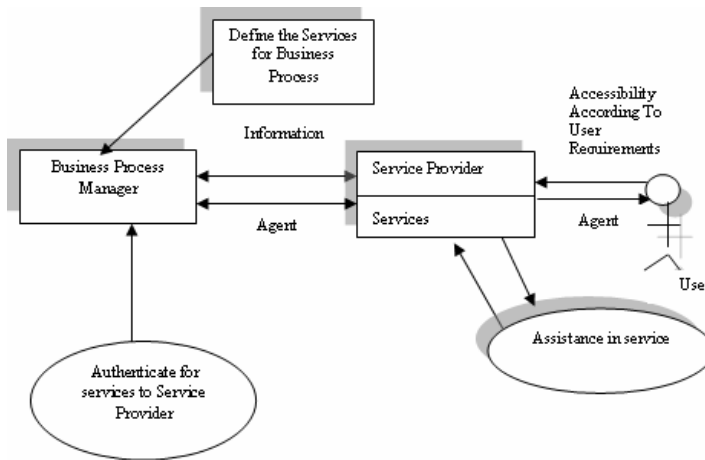
- establish Communication between the Web server and the web site
- provides Exchange of Services between the two
- transfers the data (request)

(the above task is accomplished by Communication Switching Agent)

Procedure Sequence :

- a web user on selecting a particular site establishes the connection with the web server of that site

- communication is established between the two by the Communication Switching Agent



**Figure 4:** SOA Scenario with Actors

### 5.2 AIA (*Application Interface Agent*)

Aim : to provide interface between Web server and the application

Task (action) of the agent :

- establish Interface between the web server and the application
- check for adaptive environment (operating system and application platform)
- passing of information from one website to the other as required with the help of Communication Service Agent

(the above task is accomplished by Interface Switching Agent)

Procedure Sequence :

- the web server collects the information being searched and the selected site
- an interface is established between the web server and the selected application through any adaptive environment by the Interface Switching Agent

### 5.3 DAA (*Data Adaptation Agent*)

Aim: to help exchange and passing of data between Server and the Database

Task (action) of the agent :

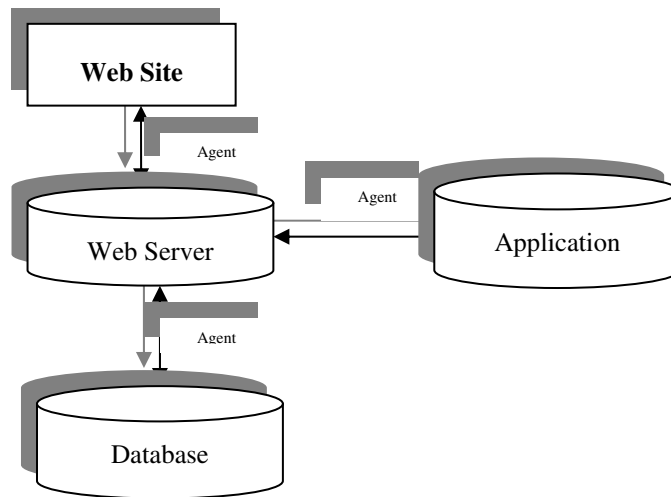
- to exchange the available / required data
- to help in adding of new data
- to modify the existing data
- to delete the unwanted data

(the above task of Updating of data is accomplished by Data Exchange Agent)

Procedure Sequence :

- the selected and required matter is made available of the huge repository available for the particular application from the data storage

- this data pupation is performed by Data Exchange Agent



**Figure 5 : Agent based SOA configuration**

In the SOA configuration of our agent based system the user's request is processed with the help of an agent which searches for the contents in the web server which in turn takes help of the agent to look for the adaption with the application being asked for and another agent which searches for the data being searched from the database storage.

Along with these agents there are few agents, which is also useful for SOA. These are as follows:

- **WEB AGENTS** : these agents act as the interface between various requestors and responders on the world wide web
- **APPLICATION AGENTS** : these agents refer to various independent applications available which can be contacted through world wide web with other heterogeneous and homogeneous applications
- **DATA RETRIEVAL AGENTS** : these are agents which serve as repository of the data a requestor is requesting for , to an application
- **COMMUNICATION AGENTS**: these are agents who help in establishing connection between various agents (viz. between web agent and application agent, application agent and data retrieval agent etc.)

## **6. Concluding Remarks**

A framework prototype of the Service Oriented Architecture for Business dynamics is currently under development where a SOA based model is being designed and developed. The implementation result shall be presented in a sequential publication.

Future work under this research will focus on the following issues:

- Design of a conceptual framework for agent based SOA, to provide decisions for the best communication between services
- Implementation of agents for SOA in distributed Environment.
- Design and development of agents for SOA
- Algorithm best agent based SOA practices



## References

1. AgentLink III. Agent Technology Roadmap. Available from <http://www.agentlink.org/roadmap/index.html>
2. Akkermans, H. (2001) *Intelligent E-Business - From Technology to Value*. IEEE Intelligent Systems, 16(4):8-10,
3. Bechhofer, R. Volz, and P. Lord. (2003) *Cooking the semantic web with the OWL API*. In Proc. Int Semantic Web Conference, pp. 659-675, Sanibel Island, FL
4. Beneventi, A., Poggi, A., Tomaiuolo, M., & Turci, P. (2004) *Integrating Rule and Agent-Based Programming to Realize Complex Systems*. WSEAS Trans. on Information Science and Applications, 1(1):422-427,
5. Bergenti, F., Poggi, A., Tomaiuolo, M., Turci, P. (2005) *An Ontology Support for Semantic Aware Agents*. In Proc. Seventh International Bi- Conference Workshop on Agent-Oriented Information Systems (AOIS- 2005 @ AAMAS), Utrecht, The Netherlands
6. Berners-Lee, T., Hendler, J., Lassila O. (2001) *The Semantic Web -A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. 284(5):34-43.
7. Buhler P.A., Vidal, J.M. (2005) *Towards Adaptive Workflow Enactment Using Multiagent Systems*. Information Technology and Management, 6(1):61-87
8. De Bruijn, J., Polleres, A., Lara, R., Fensel, D. (2005) *OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web*. In Proc. of the 14th Int. World Wide Web Conference (WWW2005), pp. 623-632, Chiba, Japan
9. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylonen, T. (1999) *SPKI Certificate Theory*. RFC 2693,
10. Fensel, D., Bussler, C. (2002) *The Web Service Modeling Framework WSMF*. Electronic Commerce Research and Applications 1(2): 113-137.
11. Forgy, C. L. (1982). *Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem*, Artificial Intelligence 19(1), pp. 17-37.
12. Greenwood, D., Callisti, M. (2004) *Engineering Web Service-Agent Integration*. In IEEE Conference of Systems, Man and Cybernetics, Available from <http://www.whitestein.com/resources/papers/ieeesmc04.pdf>.
13. Gibbins, N., Harris, S., Shadbolt, N. (2003) *Agent-based semantic web services*. In Proc of the 12th International World Wide Web Conference (WWW2003), Budapest, Hungary.
14. Gutknecht, O., Ferber, J., Michel, F. (2001) *Integrating tools and infrastructures for generic multi-agent systems*. In Proc. of the 5th International Conference on Autonomous Agents. Montreal, Canada.
15. Gutmann, P. (2000). X.509 Style Guide. Available from <http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt>
16. Horrocks, I. and Patel-Schneider, P. F. (2004) *A proposal for an OWL rules language*. In Proc. of the Thirteenth International World Wide Web Conference (WWW2004), pp. 723-731.
17. Hindriks, K.V., de Boer, F.S., van der Hoek, & W., Meyer, J.C. (1998) *Control Structures of Rule-Based Agent Languages*. In Lecture Notes In Computer Science, Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages, Vol 1555, pp. 381-396, London, UK: Springer-Verlag.
18. Khare, R., Rifkin, A. (1997) *Weaving a Web of trust*, World Wide Web Journal Special Issue on Security, 2(3):77-112
19. R. Kishore, H. Zhang & R. Ramesh, (2006) *Enterprise integration using the agent paradigm : foundations of multi-agent-based integrative business information systems*, Decision Support Systems, 42 (1), pp. 48- 78..
20. Labrou, Y. (2002) *Agents and ontologies for e-business*. Knowledge Engineering Review, 17(1):81-85,
21. Li, N., Mitchell, J.M. (2003) *RT:A Role-based Trust-management Framework*. In Proc of the Third DARPA Information Survivability Conference and Exposition (DISCEX III), pp. 201-212, Washington, D.C.
22. Negri, A., Poggi, A., Tomaiuolo, M., Turci, P., (2005) *Dynamic Grid Tasks Composition and Distribution through Agents*,. Concurrency and Computation: Practice and Experience.
23. NegriA., A. Poggi, M. Tomaiuolo, P. Turci, (2006) *Agents for e-Business Applications*, In Proc. AAMAS 2006, Hakodate, Japan
24. Papazoglou, M.P. (2001) *Agent-oriented technology in support of e-business*. Communication of ACM, 44(4):71-77,
25. Papazoglou, M.P. (2002) *The World of e-Business: Web-Services, Workflows, and Business Transactions In Lecture Notes In Computer Science, CAiSE '02/ WES '02: Revised Papers from the International Workshop*

- on Web Services, E-Business, and the Semantic Web, Vol 2512, pp. 153-173. London, UK. Springer-Verlag
26. Poggi, A., Rimassa, G., Tomaiuolo, M. (2001) *Multi-user and security support for multi-agent systems*. In Proc. of WOA 2001, pp. 13-18, Modena, Italy: Pitagora.

### ***About the Authors***

*O. P. Rishi* received the M.Tech. Degree in Computer Science from Birla Institute of Technology, Mesra Ranchi (India) in 2000, the M.Sc. Degree in Mathematics from University of Rajasthan in 1995, He has submitted his Ph. D. in Computer Science at Banasthali Vidhyapeeth, Banasthali Rajasthan in 2008. His current research involves Case Base Reasoning for Intelligent Tutoring Systems and Intelligent Systems. Presently working with Birla Institute of Technology, Mesra Ranchi, Jaipur Campus as Sr. Faculty in Computer Science Department. He has co-authored more than 10 papers in academic journals and international conferences and is a member of the editorial board/review board in a number of leading journals. He is also in the process of guest editing issues for three well respected international journals

*Ashok Sharma* received the PGDBM from IIRM Jaipur (India) in 2000, the M. Com.(Business Admin.), & M.A.(Sociology) from University of Rajasthan. Presently working with Birla Institute of Technology, Mesra Ranchi, Jaipur Campus as Assistant Administrative Officer.

*Archana Bhatnagar* received the MCA Degree in Computer Science from IGNOU, New Delhi (India) in 2000, the PGDCA. Degree in Computer Science at Banasthali Vidhyapeeth, Banasthali Rajasthan in 1990, Presently working with Birla Institute of Technology, Mesra Ranchi, Jaipur Campus as Faculty in Computer Science Department.

*Ashutosh Gupta* received the M.Tech. Degree in computer Science from Birla Institute of Technology, Mesra Ranchi (India) in 2000, the B.E. Degree in Computer Science,. Presently working with Sir Padampat Singhania University Udaipur Rajasthan as Sr. Faculty in Computer Science Department.