

Semantic Web Services and Its Approaches

Tauqeer Ahmad Usmani,
Research Scholar, Department of Computer Science,
Kumaun University, Nainital, India.
E-mail- tausmani2003@yahoo.co.in

Prof. Durgesh Pant,
Director, School of Computing & Informatics,
Uttarakhand Open University, Dehradun Campus, Uttarakhand, INDIA
(On leave from Department of Computer Science, Kumaun University, Nainital, India)
E-mail- durgesh_pant@yahoo.com

Prof. Kunwar Singh Vaisla
Department of Computer Sc. & Engineering,
BCT Kumaun Engineering College, Almora, India
E-mail- vaislaks@rediffmail.com

Abstract- OWL-S, IRS, WSMF are the prominent field that are the major part for Semantic Web Services. IRS-III is the first WSMO Compliant and implemented structure to support Semantic Web Services. IRS-III is the extension of previous version of IRS-II and supporting WSMO ontology within the IRS-III Server, browser and API. IRS-III provides support for the OWL-S service descriptions by importing the description to IRS-III. This paper describes about different approaches of Semantic Web Services.

Keyword: IRS-III, OWL-S, WSMF, WSMO, SWS

I. INTRODUCTION

Web services are software components that are developed using specific technologies from three primary technology categories:

- An XML based description format (example WSDL)
- An application messaging protocol (example SOAP) and
- A collection or transport protocol (example HTTP)

The goal of Semantic Web services is to bring the Web to its full potential. Web service technology brings a dynamic aspect to the use of the web ; on the other hand, Semantic Web technology facilitates information search, retrieval, representation, extraction, interpretation and maintenance. Today's Web service technology based on SOAP, WSDL and UDDI registry, does not capture enough data semantics or business logic. However a combination of two technologies will leverage the potential of the Web service technology. Semantic annotation facilitates service discovery and provides a more sophisticated solution to the selection, composition and interoperability across various services [3].

Semantic Web Services (SWSs) extend the idea of the Semantic Web to WS. They aim to complement the current knowledge-poor syntactic industry standards with semantic metadata in order to facilitate automation of WS related tasks such as discovery, matchmaking, and composition. The pioneering standards in this area were DAML-S and OWL-S. The most recent developments are related to WSMO, its language modeling language WSML and the reference implementation WSMX developed mostly in the scope of SWWS and DIP projects. The metadata creation problem persists also in the context of SWSs – tools to facilitate the creation of Web Service ontologies and the generations of SWS descriptions are required as the number of available services increases [7].

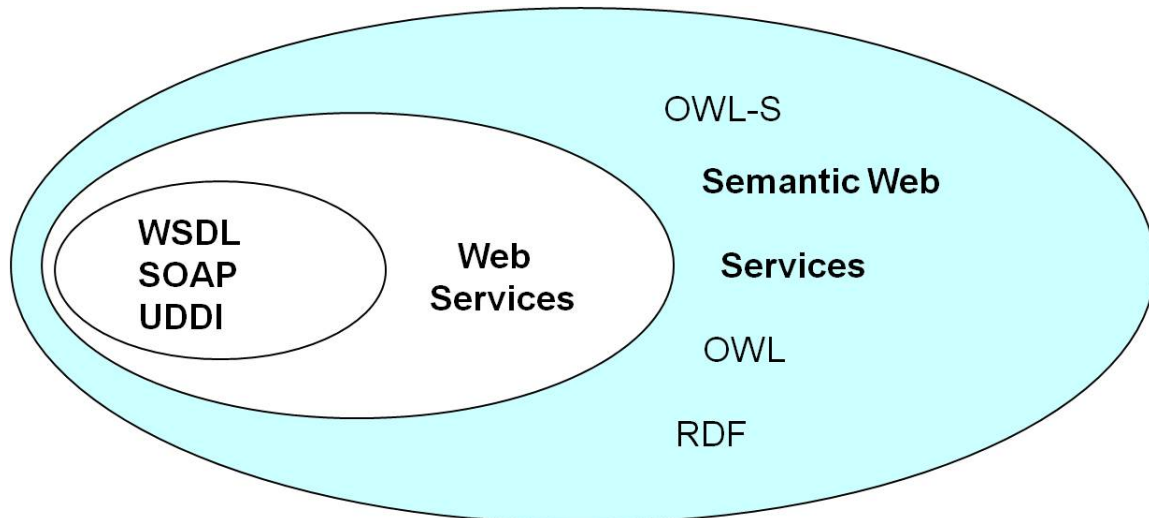


Figure 1. Semantic web Services

Current efforts in developing Semantic Web Service infrastructures can be characterized along three orthogonal dimensions: usage activities, architecture and service ontology. Usage activities define the functional requirements, which a framework for Semantic Web Services ought to support. The architecture of SWS describes the components needed for accomplishing the activities defined for SWS, whereas the service ontology aggregates all concept models related to the description of a Semantic Web Service.

There are basically three approaches that are force the growth of Semantic Web Service (SWS) framework: IRS-III, OWL-S and WSMF. Internet Reasoning Service (IRS-III) is knowledge based approach to SWS, which evolved from research on reusable knowledge component OWL-S is an agent-Oriented approach to SWS, providing fundamentally an ontology for describing Web service capabilities. Web Service Modeling Framework (WSMF) is a business –Oriented approach to SWS, focusing on a set of e-commerce requirements for web services including trust and security.

II. IRS APPROACH:

IRS-III is a framework and implemented infrastructure which supports the creation of Semantic Web Services according to the WSMO ontology and builds upon the previous version (IRS-II) by incorporating and extending the WSMO ontology within the IRS-III server, browser and API.

The Internet Reasoning Service is a Semantic Web Services framework, which allows applications to semantically describe and execute Web services. The main components of IRS-III architecture are the IRS-III Server, the IRS-III Publisher and the IRS-III Client which communicate through the SOAP protocol. The IRS-III Server holds descriptions of Semantic Web Services at two different levels. A knowledge level description is stored using the UPML framework of tasks, PSMs and domain models. IRS-III has special purpose mapping mechanism to ground competence specifications to specific Web services. The IRS-III Publisher plays two roles in the IRS-III architecture:

- It links Web services to their semantic descriptions within the IRS-III server, and then
- The publisher automatically generates a wrapper which turns the code into a Web service.

Once this code is published within the IRS-III it appears as a standard message-based Web service, that is, a Web service endpoints is automatically generated. There can be more than one type of Publisher or publishing platform, depending on the implementation of the service. a main aspect of IRS-III is that Web service invocation is capability driven.[2]

A. IRS-III Architecture:

The IRS-III architecture is consisting the main following components: the IRS-III Server, Publisher and Client, which communicate through a SOAP-based protocol as shown in figure 2. The IRS-III Server is based on an HTTP server written in LISP (a programming Language) which has been extended with a SOAP handler. Separate modules handle SOAP based requests from the browser, the publishing platforms and the invocation client. Messages result in a combination of queries to or changes within the entities stored in the WSMO library. IRS-III was designed for ease of use, in fact a key feature of IRS-III is that web service invocation is capability driven. The IRS-III Client supports this by providing a goal-specific invocation mechanism. An IRS-III user simply asks for a goal to be solved and the IRS-III broker locates an appropriate web service semantic description and then invokes the core deployed web service.

The following describes the main application development activities supported by IRS-III when building Semantic Web Services:

- **Using domain ontologies** - The concepts and relations involved in the application scenario which are used to describe client requests and Web Service capability are provided in domain ontologies.
- **Describing client requests as goals** – The request for a service can be expressed from a business viewpoint and represented as a *goal*.
- **Semantically describing deployed Web Services** – The concepts defined in domain ontologies can be used in a *web service* description to represent the types of inputs and outputs of services and in logical expressions for expressing applied restrictions. This description can also include many other aspects such as *orchestration* and *choreography*.
- **Resolving conceptual mismatches** – *Mediator* descriptions can be used to declare which *mediation service* or *mapping rules* will provide conceptual alignment between *goals*, *web services* and domain ontologies.
- **Publishing and invoking semantically described Web Services** - Once a semantic description has been created for a deployed Web Service as above, it can be registered into IRS-III for *goal*-based invocation.

The IRS-III tooling consists of a Java API and a browser /editor which support developers in building applications out of Semantic Web Services. The IRS-III browser provides an easy to use graphical interface to support the creation of WSMO descriptions, to publish deployed Web Services against these descriptions and then to invoke the Web Services. The IRS-III Java API provides a data model for our WSMO implementation and remote access to the operations available from the IRS-III server. Recently, we have also developed a plug-in for WSMO Studio [4] for interoperability purposes, by aligning the IRS-III and WSMO4J (<http://wsmo4j.sourceforge.net>) APIs.

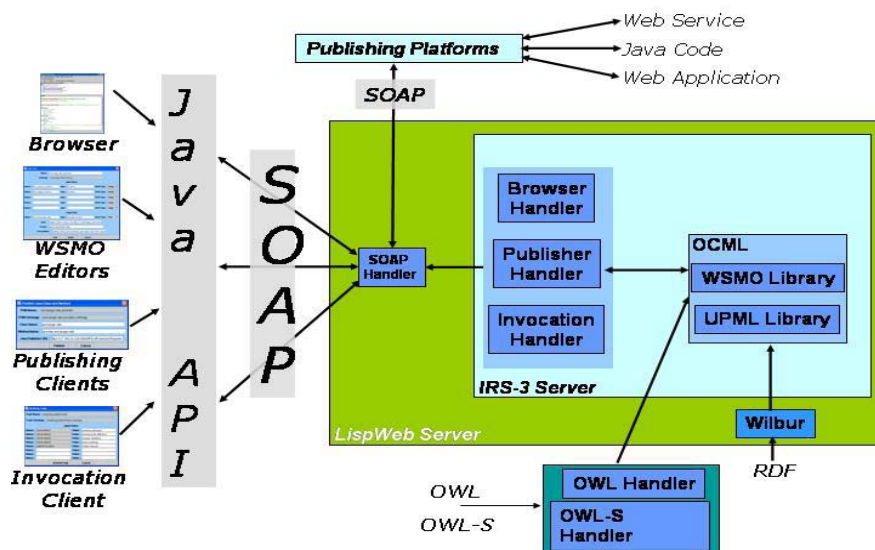


Figure 2. The IRS-III Server Architecture

IRS-III provides the representational and reasoning mechanisms for implementing the WSMO meta-model in order to describe Web Services. Additionally, IRS-III provides a powerful execution environment which enables these descriptions to be associated to a deployed Web Service and instantiated during selection, composition, mediation and invocation activities [11].

B. The IRS-III Service Ontology

The IRS-III service ontology has originally been based on the UPML framework [14] [13], which form the epistemological basis for IRS-III. This framework has been extended in order to incorporate the following main aspects specified by the WSMO conceptual model [15]:

- **Non-functional properties** – These properties are associated with every main WSMO element and can range from information about the provider such as organization, to information about the service such as category, cost or trust, to execution requirements such as scalability, security or robustness.

- **Goal-related information** – a *goal* represents the user perspective of the required functional capabilities. It includes a description of the requested *web service capability*.
- **Web Service functional capabilities** – Represent the provider perspective of what the service does in terms of inputs, output, pre-conditions and post-conditions. Preconditions and post-conditions are expressed by logical expressions that constrain the state or the type of inputs and outputs.
- **Choreography** – The *choreography* specifies how to communicate with a Web Service. In WSMO this specification is formalized as Abstract State Machines.
- **Grounding** – The grounding is associated with the *web service choreography* and describes how the semantic declarations are mapped to a syntactic specification such as WSDL.
- **Orchestration** – The *orchestration* of a *web service* specifies the decomposition of its capability in terms of the functionality of other Web Services. In WSMO this specification is also formalized as Abstract State Machines.
- **Mediators** – In WSMO, a *mediator* defines which WSMO top elements are connected and which type of mismatches can be resolved between them.

The IRS-III implementation of the WSMO conceptual model has been extended in the following ways.

- **Explicit input and output role declaration** – IRS-II I requires that *goals* and *web services* have input and output roles, which include a name and a semantic type. The declared types are imported from domain ontologies.
- **Web Services are linked to Goals via mediators** - If a *wg-mediator* associated with a *web service* has a *goal* as a source, then this *web service* is considered to solve that *goal*. An *assumption* expression can be introduced for further re-financing the applicability of the *web service*.
- **GG-mediators provide data-flow between sub-goals** In IRS-III, *gg-mediators* are used to link *sub-goals* within an *orchestration*, and therefore they can provide dataflow and data mediation between the *sub-goals*.
- **Web Service can inherit from Goals** - *Web services* which are linked to *goals* ‘inherit’ the *goal’s* input and output roles. This means that input role declarations within a *web service* are not mandatory and can be used to either add extra input roles or to change an input role type.
- **Client Choreography** – The provider of a *web service* must describe the *choreography* from the viewpoint of the client. This means IRS-III can interpret the choreography in order to communicate with the deployed Web Service.
- **Mediation services are goals** – A *mediator* can declare a *goal* as the *mediation service* which can simply be invoked. The associated *web service* actually performs the necessary data transformation [16].

III. OWL-S APPROACH:

OWL-S represents an upper ontology for the description of Semantic Web Services expressed in OWL. OWL-S consists of a set of ontologies designed for describing and reasoning over service descriptions. OWL-S approach originated from an Artificial Intelligence setting. It has earlier been used to describe agent functionality within several Multi Agent Systems as well as with a variety of planners to solve higher level goals.

OWL-S (previously DAML-S [8]) consists of a set of ontologies designed for describing and reasoning over service descriptions. OWL-S approach originated from an AI background and has previously been used to describe agent functionality within several Multi-Agent Systems as well as with a variety of planners to solve higher level goals. OWL-S combines the expressivity of description logics (in this case OWL) and the pragmatism found in the emerging Web Services Standards, to describe services that can be expressed semantically, and yet grounded within a well defined data typing formalism. It consists of three main upper ontologies: the Profile, Process Model and Grounding. The Profile is used to describe services for the purposes of discovery; service descriptions (and queries) are constructed from a description of functional properties (i.e. inputs, outputs, preconditions, and effects - IOPEs), and non-functional properties (human oriented properties such as service name, etc, and parameters for defining additional meta data about the service itself, such as concept type or quality of service). In addition, the profile class can be subclassed and specialized, thus supporting the creation of profile taxonomies which subsequently describe different classes of services.

OWL-S complements industry efforts such as SOAP, WSDL, UDDI, and BPEL4WS. It builds upon these efforts by adding rich typing and class information that can be used to describe and constrain the range of Web service capabilities much more effectively than XML data types. Further, it integrates such rich class representations with a process model, designed not only to capture the control flow and data flow of Web services, but also their side effects (preconditions and effects) in the world. The use of such a language enables the grouping of like services and like data types into taxonomic hierarchies, together with rich definitions of the relationships and constraints between classes and their instances. The well-defined semantics enables automated manipulation of these structures, with known outcome. In short, OWL-S makes automated interoperation possible [9].

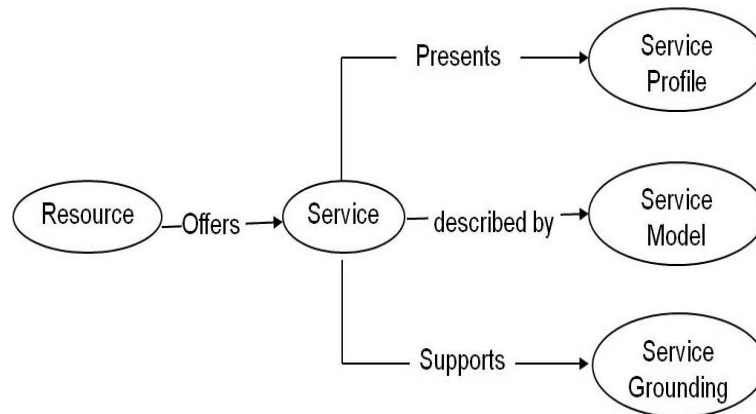


Figure 3. OWL-S service

OWL-S blends the expressivities of description logics and the simplicity established in the emerging Web Services Standard, to describe service that can be expressed semantically. OWL-S is originated around three interrelated sub-ontologies: ServiceProfile, ServiceModel and ServiceGrounding, as shown in figure 3. The ServiceProfile describes 'what the service does'. This description is essential if an agent is to determine whether or not the service meets its need. The ServiceModel explains 'how service works'. This is a process model which describes how to use the service, how to ask for it and what happens when the service is carried out. The ServiceGrounding gives the details of how a service requester agent can interact with a service. These details can be expressed in WSDL specifying a communication protocol, message formats and other service related details.. Additionally, the ServiceGrounding must specify for each semantic type of input or output specified in the ServiceModel.

A. ServiceProfile

A service profile provides a high-level description of a service and its provider and is used by discovery registries to request and advertise services. It includes: a human readable description, a specification of functionalities, and functional attributes. A transaction in the Web service marketplace involves three parties: the service requester, the service provider, and the infrastructure components. The service requester (buyer) seeks a service to complete its work and the service provider (seller) offers a service. The infrastructure components facilitate the process, such as registries to match the request with the offers available. Within the OWL-S ontology, the ServiceProfile describes the services. A ServiceProfile describes a service as a function of three basic types of information: what organization provides the service, what functions the service computes, and what features characterize the service. There is a two-way relationship between a service and a profile which is expressed by the properties presents (relates an instance of service and an instance of profile) and presented by (specifies the inverse of presents).

B. ServiceModel

A detailed perspective of a service can be viewed as a process. A subclass of the ServiceModel is defined as the ProcessModel which draws upon Artificial Intelligence, planning and workflow automation to support a wide array of services on the Web. Figure 4 presents the relationships of ServiceModel and includes the Process and Process Control.

C. ServiceGrounding

In OWL-S, both the ServiceProfile and the ServiceModel are thought of as abstract representations; only the ServiceGrounding deals with the concrete level of specification. The grounding of a service specifies the details of how to access the service, including protocol and message formats, serialization, transport, and addressing [10]

IV. WSMF APPROACH

The Web Service Modeling Framework (WSMF) provides a model for describing the diverse aspects related to Web services. Its main purpose is to make e-commerce possible by applying Semantic Web technology to Web services. WSMF is the product of research on modeling of reusable knowledge components. WSMF is based on:

- a strong decoupling of the various components that realize an e-commerce application; and
- a strong mediation service enabling Web services to communicate in a scalable manner.

Moreover, WSMF consists of four main elements

- Ontologies that offer the terminology used by other elements;
- target repositories that define the problems to be tackled by Web services;
- Web service metaphors that define facets of a Web service; and
- mediators who bypass interoperability problems.[2]

WSMF is centered on two complementary principles: a strong de-coupling of the various components that realize an e-commerce application; and a strong mediation service enabling Web services to communicate in a scalable manner. Mediation is applied at several levels: mediation of data structures; mediation of business logics; mediation of message exchange protocols; and mediation of dynamic service invocation.

WSMO service ontology includes definitions for goals, mediators and web services. A web service consists of a capability and an interface. The underlying representation language for WSMO is F-logic. The rationale for the choice of F-logic is that it is a full first order logic language that provides second order syntax while staying in the first order logic semantics, and has minimal model semantics. The main characterizing feature of the WSMO architecture is that the goal, web service and ontology components are linked by four types of mediators as follows:

- OO mediators link ontologies to ontologies,
- WW mediators link web services to web services,
- WG mediators link web services to goals, and finally,
- GG mediators link goals to goals.

Since within WSMO all interoperability aspects are concentrated in mediators the provision of different classes of mediators based on the types of components connected facilitates a clean separation of the different mediation functionalities required when creating WSMO based applications [12].

V. DISCUSSION AND CONCLUSION:

SWSs are commonly seen as building blocks for processes that can be rearranged by business experts without involving developers. SWS research has been receiving increasing attention during the past years. Currently several efforts are involved in SWS technology research and development, mostly gathered around OWL-S (Web Ontology Language for Services) and WSMO (Web Service Modeling Ontology), which are the most well known SWS frameworks. Basic knowledge about Web services and the Semantic Web as well as insight into the things that most SWS frameworks have in common is particularly valuable when discussing the relevance and applicability of SWS-based integration architectures.

There are some analogy between OWL-S profiles and WSMO capabilities; this does not indicate that OWL-S has a counterpart to goal descriptions as found in IRS-III.

OWL-S lacks a holistic conceptual model which could facilitate a rapid and consequent realization of Semantic Web Services. A vertical prototype which demonstrates techniques for service discovery and execution for own implementations would be highly desirable.

For a successful discovery, the publisher of service registers the service at a registry and describes its capabilities. To discover a service, a requester must specify a goal and translate it into a machine readable query. The goal is then decomposed into constituent sub goals and matched against the capabilities of the registered services. We propose the machine readable query from human readable query should be optimal and automated so that exact or desired search domain will minimize.

WSMO based approaches are currently the most powerful due to better performance characteristics, heterogeneity support and their holistic approach towards service discovery.

References:

- [1]. John Hbeler, Matthew Fisher, Semantic Web Programming, Wiley India Pvt. Ltd. ISBN: 978-81-265-2110-4, pg.-97
- [2]. Rajendra Akerker, Foundation of Semantic Web, Narosa Publishing House, India, ISBN 978-81-7319-985-1 pg-132, 133
- [3]. Karin K.Breitman, Springer International Edition, ISBN 978-81-8489-397-7 pg 127-128.
- [4]. F.Hakimpour, J. Domingue, "Integration of OWL-S into IRS- II", AKT Workshop on Semantic Web Services, AKT- SWS04,2004
- [5]. <http://www.w3.org/2005/04/FSWS/>
- [6]. Cabral L, Domingue, J ; Motta E;Hakimpour F " Approaches to Semantic Web services: an overview and comparison, Proceedings of the European Semantic Web Conference, 2004
- [7]. <http://www.tao-project.eu/researchanddevelopment/technologies/sws.html>
- [8]. Fensel, D., Bussler, C. The Web Service Modeling Framework WSMF. Eletronic Commerce: Research and Applications. Vol. 1. (2002). 113-137
- [9]. Katia Sycara, David Martin, "OWL-S Technology for Representing Constraints and Capabilities of Web Services". <http://www.w3.org/2004/08/ws-cc/dmowls-20040904>
- [10]. Alesso, H. Peter Smith, Craig F., Developing Semantic Web Services, Publisher: A K Peters, Limited, ISBN: 9781568812120, pg-279, 286
- [11]. John Domingue, Liliana Cabral, Farshad Hakimpour, "Demo of IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services", Knowledge Media Institute, The Open University,
- [12]. Liliana Cabral, John Domingue, "Approaches to Semantic Web Services", Knowledge Media Institute, The Open University, Milton Keynes, UK
- [13]. Motta, E., Domingue, J., Cabral, L. and Gaspari, M.: IRS-II: A Framework and Infrastruc-ture for Semantic Web Services. In proceeding of the 2nd International Semantic Web Con-ference (ISWC 2003). LNCS 2870 (2003)
- [14]. Omelayenko, B., Crubezy, M., Fensel, D., Benjamins, R., Wielinga, B., Motta, E., Musen, M., Ding, Y.: UPML: The language and Tool Support for Making the Semantic Web Alive. In: Fensel, D. et al. (eds.): Spinning the Semantic Web: Bringing the WWW to its Full Potential. MIT Press (2003)141–170
- [15]. WSMO Working Group. Deliverable D2v1.2 Web Service Modeling Ontology (WSMO). <http://www.wsmo.org/TR/d2/v1.2/> (2005)
- [16]. Liliana Cabral, John Domingue, Stefania Galizia, Alessio Gugliotta, "IRS-III: A Broker for Semantic Web Services based Applications", Knowledge Media Institute, The Open University, Milton Keynes, UK

AUTHORS PROFILE



Tauqeer Ahmad Usmani received Bachelor of Science B.Sc.(Hons.) and Master(MCA) degrees in Computer Application from L N Mithila University and Magadh University in 1995 and 2000 respectively. Currently pursuing Ph.D. from Kumaun University, Nainital. Presently working as Lecturer, Salalah College of Technology(Ministry of Manpower), Sultanate of Oman. Having 11 years of teaching experience in higher education including India and abroad. Published paper in International Journal. Members of various professional bodies of India and international repute.The research areas are Semantic Web, Intelligent Web.



Durgesh Pant received the Graduation in Science (B. Sc.) from Kumaon University, Nainital, Master (MCA) degrees in Computer Applications from BIT, Mesra, Ranchi, and Ph. D. in Computer Science from BIT Mesra, Ranchi. Working as Professor, Computer Science, Kumaon University, Nainital, since 20 years and presently working as Director, Dehradun Campus, Uttarakhand Open University, Uttarakhand. He started computer science in the Kumaon University, Nainital. He has guided / supervised more that 15 Research students. Interested field of research are ICT impact on G2C of e-Governance, Data Warehouse and Mining, IBIR. Authored many research papers in International / national journals/conferences in the field of computer science and also many books in reputed publishing house.



Kunwar Singh Vaisla received the Graduation in Science (B. Sc.) and Master (MCA) degrees in Computer Applications from University of Rajasthan, Jaipur in 1994 and 1998, respectively. Recently (in Jan 2011) completed Ph. D. in Computer Science from Kumaon University, Nainital. Presently working as Associate Professor (Computer Science & Engineering) in Kumaon Engineering College (A Govt. Autonomous College), Dwarahat (Almora) – Uttarakhand. Interested field of research are ICT impact on G2C of e-Governance, Data Warehouse and Mining, Complex / Compound Object Mining, IBIR. Authored many research papers in International / national journals/conferences in the field of computer science and also many books in reputed publishing house.