# ENABLING COLLABORATION IN THE SEMANTIC GRID: SURVEY OF WEB SERVICE COMPOSITION APPROACHES

Taha Osman, Dhavalkumar Thakker, David Al-Dabass
School of Computing & Informatics,
Nottingham Trent University,
CIB, Nottingham NG11 8NS, United Kingdom.
Email: taha.osman@ntu.ac.uk, dhavalkumar.thakker@students.ntu.ac.uk, david.al-dabass@ntu.ac.uk

**KEY WORDS**

Web Services, Service Composition, Semantic Web, Business Process Management.

**ABSTRACT**

Web Service is loosely coupled highly accessible distributed computing technology that can expose applications beyond the firewall. Composition of Web Services has received much attention from the business and the research community. Composition techniques are classified as static, dynamic and semi-automatic composition, each addressing different application areas and requirements. In this contribution we analyze workflow-based and semantic-based composition approaches, primarily focusing on the facilitation to the service participants and the scalability required in a Grid-based environment.

## 1. INTRODUCTION

The last decade has witnessed an explosion of application services delivered electronically, ranging from e-commerce to information service delivered through the World Wide Web (WWW) to the services that facilitate trading between business partners, better known as Business-to-Business (B2B) relationships. Traditionally these services are facilitated by distributed technologies such as RPC, CORBA and more recently RMI. Web Services is the latest distributed computing technology. It is a form of remote procedure call like other distributed computing technology, but uses XML extensively for the messaging, discovery and description. The use of XML messaging makes Web Services platform and language neutral. Web Services use SOAP (Simple Object Access Protocol) for XML messaging, which in turn uses ubiquitous HTTP for the transport mechanism. HTTP is considered as a secure protocol thus it allows the Web Services to be exposed beyond the firewall. The Web Service messages and operations with invocation details are described using a platform-independent language WSDL (Web Services Description Language). Web Services can be published and discovered using UDDI (Universal Description Discovery and Integration) protocol. The Web Services architecture centred on WSDL, UDDI and SOAP is an instance of Service Oriented Architecture (SOA). Using this architecture services can be published using UDDI, with WSDL based description, and can be searched, called and bind at run time making it loosely coupled and highly accessible.

To take advantage of these features of Web Services, network applications services have to be developed as Web Services or converted into Web Service using the wrapping mechanism (Osman T et al, 2005). Moreover, multiple Web Services can be integrated either to provide a new, value-added service to the end-user or to facilitate co-operation between various business partners. This integration of Web Services is called "Web Services composition" and is feasible to achieve because of the Web Services advantages of being platform, language neutral and loosely coupled. The composition is particularly apt for grid environments, where internet-wide computing resources are available for application services to interoperate and collaborate.

The logic for the composition mainly involves two activities: selection of the candidate Web Services that fulfil the requirement in accumulation and flow management. Flow management is further categorized into control and data flow, where control flow is the order in which Web Services operations are invoked, while the data flow is the order in which the messages are passed between the Web Services operations. The level of automation provided in performing selection of services and flow management classifies composition into static, semi-automatic and dynamic. Static composition involves prior hard coding of the service selection and flow management. Performing selection and flow management on the fly, in machine-readable format leads to dynamic composition. In semi-automatic composition, service composer is involved at some stage.

This study shows that these approaches can be divided into two categories. The first category largely endorsed by the industry, borrows from business processes' workflow management theory to achieve the formalization necessary for describing the data flow and control flow in the composition scheme. The second category mainly promoted by the research community, aspires to achieve dynamic composition by semantically describing the process model of Web service and thus making it comprehensible to reasoning engines or software agents.

Proceedings of the 2008 High Performance
Computing & Simulation Conference ©ECMS
Waleed W. Smari (Ed.)
ISBN: 978-0-9553018-7-2 / ISBN: 978-0-9553018-6-5 (CD)

The structure of the paper is as follows: sections 2 and 3 discuss workflow-based and semantic-based composition techniques respectively. Section 4 provides evaluation of the surveyed composition techniques and in section 5 we conclude the paper.

## 2. WORKFLOW MANAGEMENT THEORY-BASED APPROACHES

Workflow is the movement of documents and/or tasks through a work process. More specifically, workflow is the operational aspect of a work procedure: how tasks are structured, who performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked (van der Aalst 2003).

Workflow management systems are a class of information systems that make it possible to correlate people's work and computer applications. Such systems deal with the control flow (invocation sequence of applications) and data flow (information flow between applications) while control flow is important for achieving overall system objective, data flow is essential for the successful operation of individual applications.

In the information systems domain, workflow is being used since seventies for the office automation systems (Zisman 1977). This work has lead to identifications of workflow patterns for control and data flow (van der Aalst 2003).

One of the applications of workflow management in information systems domain is to address the Business Process Management (BPM) problem. Business process can be considered as workflow of business activities to carry out business goals (Leymann 2002). The examples of business activities for customer order fulfilment business process are: customer placing an order, checking account status, verifying order and despatch. Using Workflow management, BPM deals with achieving the integration of these individual applications.

Business process can have scope within inter and intra organization relations. Enterprise Application Integration (EAI) is the BPM solution to achieve intra-organization business applications integration, while Business-to-Business (B2B) integration software addresses the problem for inter organization business application integration. Traditional EAI and B2B integration solutions are very complex, proprietary and presume many details about the participating applications making them tightly coupled. For instance, these solutions assume the use of homogeneous service interfaces and implementation technology, which is a substantial limitation considering that different organizations will make independent decisions about what technology to use for the construction and deployment of their part; these decisions made over time accrete different hardware and software technologies (High 2004). Tightly coupled systems are difficult to manage and re-engineering business rules and requirements in such systems is also challenging. To overcome these limitations, the business applications are now being developed using Web services while the BPM problems (EAI, B2B) are being addressed with the workflow based integration of Web services, mainly to utilize SOA based Web services features.

The main industrial standards to achieve workflow based integration of Web services are WS-BPEL (Web Services Business Process Execution Language, shortened to BPEL) (Andrews 2003), WS-CDL (Web Services Choreography Description Language, shortened to CDL) (Kavantzas 2004) and BPML (Business Process Modelling Language) (Van der Aalst 2003). These approaches use WSDL extensively and build workflow based on WSDL operations and messages with the data types. The workflow based process model for these approaches also addresses requirements for describing flow management in composition, handling business transaction with roll back facility, state management for business interaction support, and also handling exception and errors. The category of process model and the extent to which these features are provided differentiates these standards.

The following sections outline two prominent workflow based industrial standards for Web services composition.

### 2.1 Composing services using BPEL

The BPEL specification - enhances and replaces existing standards Web Services for Business Process Design (XLANG) (Thatte 2001) from Microsoft and Web services Flow Language (WSFL) (Leymann 2002) from IBM. The specification uses workflow management as process model to achieve the control and data flow formalization for WSDL defined data and operations. All the participant services in BPEL process are modelled as partners. The WSDL files of such partners are required to create BPEL process. The partners contribute to the total processing capability of the BPEL process. BPEL process also has its own processing capability for dataflow, control flow, data manipulation, fault and event handling and state management. The significance of BPEL architecture is that the process itself is published as a Web Service. This composed BPEL service can be treated as a single Web service and can be used for further composition hence facilitating recursive composition.

### 2.2 Composition using WS-CDL

BPEL process model deals with B2B integration from a single party viewpoint i.e., the requirement specified for the travel agent scenario discussed here is from the viewpoint of travel agent business logic. Contrary to the BPEL process model, real world B2B integrations are peer-to-peer in place of being centralized, where the collaborating business applications agree to provide certain functionality in receipt of complimentary functionality from other business applications highlighting requirement for a description language documenting peer-to-peer viewpoint since natural B2B integrations are peer-to-peer collaborative relationships and not governed by a single party. The W3C recommendation WS-CDL from W3C Web services choreography working group confirms aforementioned conclusions that more work on BPEL is required to make it adoptable for B2B integration.

WS-CDL is a description language using which the B2B integration partners can first describe the collaborative functionality. This description document is considered as a contract and each party can implement their own part. The WS-CDL document describes common and complementary behaviour of all the parties involved, making the viewpoint global and peer-to-peer (Kavantzas 2004). The other aspect of WS-CDL process model is that the internal business logic of each party remains hidden from the business partners. i.e., for the travel agent application after receiving price quote from all airlines can have internal business logic for air line selection based on some criteria totally hidden from other partners as the external detail described in WS-CDL document is just an operation to make reservation at particular airline.

### 2.3 Facilitation provided to the service participants

In order to evaluate the facilitation provided to the service participants we consider a scenario based on travel agent service, which manages the reservation of airline and hotel for a customer trip. The travel agent can be implemented as BPEL process, which cans becomposition of four Web services: AirFrance service, AirUSA service, HotelRating service and HotelService service. The process logic for the travel agent is: to check the availability of flight service from two competing airlines AirFrance and AirUSA, make flight reservation, and then retrieve hotel ratings from the HotelRating service at the destination city and make the reservation using HotelService Web service at the selected hotel.

For a new service provider to make their service available for composition they need to provide minimum functionality consistent with the business logic outlined by the travel agent which is essentially composer. Considering a new AirUK service for travel agent composition, AirUK has the following options:

a) If the AirUK has Web service but does not implement required functionality then the service needs to be modified to accommodate the required functionality.

b) If the AirUK has a non-Web service application with the required functionality already built-in then only a WSDL file is required to be created without modifying existing application. BPEL execution engine uses Web Services Invocation Framework (WSIF 2005) for the Invocation of such non web-services.

c) However, BPEL specification does not address a case where the Web service provider has a service available with conceptually similar but syntactically different parameter structure. The service provider in this case needs to apply option (a) to be part of the composition.

Considering the case of service composer who for the most part encounter problems in parameter mismatch during the flow management, i.e., a service operation has different output format from the input of next service operation in the flow logic, BPEL in its current form delegate the responsibly with the service composer to address such parameter mismatch.

The travel agent BPEL process could be published using JSP technology. This way the service can be retrieved using simple web page or WSDL file for the composed Web service can be retrieved from the public UDDI registry. In such B2C interactions it is totally transparent from the end-user that the service is a Web service with the possibility of composition of multiple Web services or could be implemented on heterogeneous platforms using heterogeneous programming languages. However, there is a limited level of language expressiveness available to the service requestor to outline the constraints and preferences on the outputs and quality of service parameters.

To conclude this section, BPEL is widely-used specification for composing intra-organization Web services. The business analysts and developers can collaborate and can compose enterprise Web services manually using BPEL. The composition is hard coded and the developers should have the explicit knowledge of all the details of participating business services which is a major limitation considering the growth of Web services within and outside organizations.

## 3. SEMANTIC WEB -BASED COMPOSITION

Commercial institutions are focusing their efforts on standardizing the static composition techniques in preparation for their wider adoption amongst the business community. In contrast, the research community efforts concentrate on exploiting semantic web for the semi-automatic and automatic composition of Web services.

### 3.1 Semantic Web services

With respect to automation, the limitation of workflow-based approaches is that they rely on WSDL based description for the Web services selection. WSDL is a static interface and XML grammar which has no notion of machine interpretable semantics. In Web services protocol stack, the task of meaningful Web services discovery was the functionality of UDDI implementations so that service provider can describe the capability of their service using the WSDL descriptions and service requester can use these descriptions to retrieve exactly what they are looking for. The search in UDDI is based on keywords and based on human readable descriptions in WSDL, leaving the selection based on the requestor's interpretation and ultimately the solution static.

The problem of automatic Web services discovery and integration can benefit from the semantic web machine readable descriptions. The fundamental premise of the semantic web is to extend Web's currently human-oriented interface to a format that is comprehensible to software programmes. Applied to Web services composition, this can lead to the automation of services selection and execution.

The WSDL file of Web services describe the operations provided; request message format required for invoking operations and the format of response messages produced by the Web services. The interpretation of these details results in understanding of the service capability. The automation required for the service composition can be achieved by describing the WSDL elements semantically, thus allowing software agents to reason about the service capability, and make all the decisions related to the composition on behalf of the user or developer. The decisions include the selection of appropriate services, their actual composition and close examination of how they meet the criteria specified by the user. In contrast, in the static composition approach, the user or developer manually interprets the requirements for the required composition and the available service capability or functionality and makes decisions regarding how services can be interweaved to make a value-added service.

The WSDL specification is part of the base Web services protocol stack and has been already widely accepted and implemented to describe Web services. Considering this, the general scenario will be to annotate individual WSDL elements with corresponding OWL elements. OWL-S (Dean 2005) is such ontology specification for describing Web services semantically. OWL-S ontology provides a mechanism to describe the capability of Web services in machine-readable form, which makes it possible to discover and integrate Web services automatically. OWL-S defines three interrelated subontologies, known as the profile, process model and grounding. In brief, the profile is used to express "what a service does", for the purpose of advertising, constructing service requests and matchmaking; the process model describes "how it works", to enable invocation and composition; and the grounding maps the constructs of the process model onto detailed specifications of message formats, protocols and so forth (Martin 2004). Figure 1 outlines these subontologies.

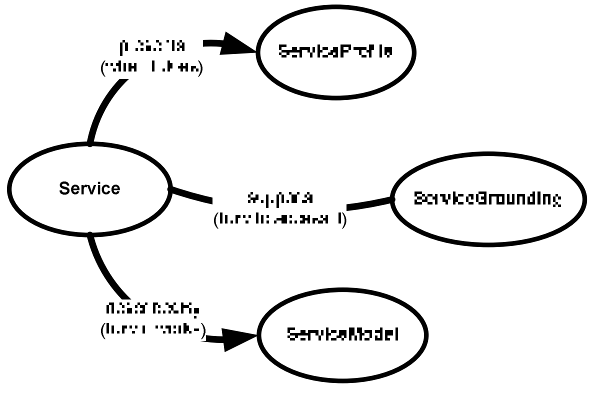


**Figure 1. OWL-S subontologies**

The OWL-S based approach facilitates the meaningful searches with the advantage of (IOPE) in profile and process based service model hence user can perform in-depth analysis of multiple services to perform a specific task.

### 3.2 Reasoning of the Service Semantics

Ontology based descriptions provides a mechanism to describe Web services functionality and the information useful for composition to be encoded in unambiguous machine understandable form. In order to perform the automated composition, an intelligent layer is essential that can interpret semantic descriptions and can order, combine and execute Web services to achieve the desired functionality or user goals. In other words, the intelligent layer should comprehend the descriptions in order to decide the possible services and build flow management for those services.

The semantics based approaches can be categorized based on the intelligent layer employed to achieve Web services discovery and composition. AI planning and case based reasoning are some of the methodologies employed as intelligent layer.

#### Artificial Intelligence Planning

This section discusses the relevancy of AI planning for the Web services composition problem and presents the literature survey on the subject.

Planning is a task of discovering a sequence of actions that can achieve a goal (Russell 2003). A planning problem can be described as a five-Tuple problem ***( S,s0,G,A,T)*** where ***S*** is the set of all possible states of the world, ***s0*** denotes the initial state of the planner, ***G*** denotes the set of goal states the planning system should attempt to reach, ***A*** is the set of actions the planner can perform in attempting to reach a goal state, and the transition relation ***T*** defines the semantics of each action by describing the state (or set of possible states if the operation is non-deterministic) that results when a particular action is executed in a given world state.

Web services composition is similar to planning problem evident from the following mapping.

***S*** is the set of possible Web services, i.e. Web services available from the service registry
***s0*** is the initial state where some or none services are pre-selected for composition
***G*** is the composition of Web services which satisfies the user requirements.
***A*** is the Web services operations (I) or preconditions (P) available to planner to reach from the initial to goal state
***T*** is the outputs (O) and effects (E) of invoking Web services operations.
AI planning dependent approaches use IOPE based OWL- S profile and process model to achieve required automation for the Web services composition. For example, if one starts with composition as goal (some desired outputs and effects), and matches it to the outputs and effects of a Web service (modelled as process), the result is an instantiation of the process, plus descriptions of new goals to be satisfied based on the inputs and preconditions of that process. The new goals (inputs and preconditions) then naturally match other processes (outputs and effects), so that composition arises naturally(Martin 2004).

Consistent with the above theory, Wu *et al* (Wu 2003) utilize DAML-S based descriptions, the previous version of OWL-S with SHOP2 planner (Kuter 2005). The SHOP2 is a Hierarchical Task Network (HTN) planner that creates plan by task decomposition - a process in which the planning system decomposes tasks into smaller and smaller subtasks, until primitive tasks are found that can be performed directly. The authors stress similarity between the concepts of task decomposition in HTN with the process decomposition in DAML-S.

Sirin *et al* in (Sirin 2004) describe another approach which couples OWL reasoner with AI planner to reason about the world state (effects and pre-condition) during planning. The reasoning is achieved by describing pre-condition and effects of the Web services using OWL.

**Case Based Reasoning**

Experience based learning using CBR is a relatively old branch of artificial intelligence and cognitive Science and is being used (Hammond 1986) as an alternative to rule-based expert system for the problem domains, which have knowledge captured in terms of experiences rather than rules. Case based reasoning for Web services were initially documented in (Limthanmaphon 2003), where the developed framework uses CBR for Web services composition. In their approach, the algorithm for Web services discovery and matchmaking is keyword based and has no notion for semantics. This affects the automation aspects for Web services search and later for composition. Similarly approach described in (Diaz 2006) proposes an extension of UDDI model for web services discovery using category-exemplar type of CBR, where web services are categorized in domains and stored as exemplar (Porter 1986) of particular domain. Their implementation of CBR reasoner facilitates UDDI registry by indexing the cases based on the functional characteristics of Web services. However, the approach does not take into consideration the importance of non-functional parameters in service selection and the use of semantics at CBR level is peripheral as they primarily use the UDDI based component for service discovery. UDDI is text-based leaving little scope for automation.

There is also a number of existing approaches which applies CBR for workflow modelling. (Madhusudan 2004) proposes an approach to support workflow modelling and design by adapting workflow cases from a repository of process models where workflow schemas are represented as cases and are stored in case repositories. The cases are retrieved for a problem which requires similar business process to solve the problem. The description and implementation language of framework is based on XML and main focus is on assisting workflow designer in creating business process flows. In similar line, (Cardoso 2005) represents adaptive workflow management system based on CBR and targets highly adaptive systems that can react themselves to different business and organization settings. The adaptation is achieved through the CBR based exception handling, where the CBR system is used to derive an acceptable exception handler. The system has the ability to adapt itself over time, based on knowledge acquired about past execution experiences that will help solve new problems.

### 3.3 Potential Facilitation to the composition participants

Despite the enthusiasm of the research community about the semantic web, there is still some way to go for creating a unifying framework facilitating the interoperation of intelligent agents or reasoning engines attempting to make sense of semantic Web services. However the workflow based approaches address here-and-now practical problem of Web services composition while dynamic Web services composition approaches holds better futuristic potential that can serve a great range of business domains. Automatic Web services composition has the potential to reduce development time and effort for the development of new applications. This is due to automatic re-configuration of changing or unavailable services in the integration.

Semantics assisted dynamic composition can serve all business domains for the possible B2B, EAI and B2C integrations. User can specify parameters for the successful composition and the composition can be performed at the run-time. The automatic Web services composition solution can address the problems of identifying candidate services, composing them, and verifying closely that they satisfy the request.

The service providers will be able to participate in the composition to their benefit with minimal effort as the development effort will be essentially reduced. The human developer will be taken out of the loop.

## 4. EVALUATION

For our research objectives, we have chosen the following criteria to study existing Web services composition approaches.

1. Service matchmaking

Using this evaluation criterion we compare various approaches based on how the service matchmaking is performed. The possible options are discovery using WSDL, UDDI, free-text or OWL-S (previously DAML-S) profile and process.

Workflow-based approaches use WSDL files to interpret the capability of a service coupled with the communications with the service provider or manual analysis of service parameters. AI planning, CSP, and agent-based approaches use different algorithms that utilize semantic web services profiles to match-make with semantically-encoded problem requests. CBR based approaches are so fare using UDDI to match-make web services.

2. Composition

We use this criterion to compare existing approaches to evaluate them based on the how they employ intelligent layers to achieve composition of Web services.

Workflow-based approaches use web services workflow languages such as BPEL and WS-CDL to outline the workflow of Web services. AI planning-based approaches utilize AI planner to form composition plans using existing planners such as SHOP2 (Levesque 1997) or GOLOG (Kuter 2005). CSP based approaches utilize existing standards WSDL, UDDI and BPEL to achieve the required composition. CBR based approaches use bespoke XML based workflow languages to write composition schema. Agents-based approaches model web services as agents so that the problem of web services composition translates to agent collaboration problem so that it is possible to utilize existing agent-infrastructure for composition.

3. Automation

Automation criterion is used to measure the level of automation achieved by various Web services composition approaches. We measure this using level of automation in the process of service discovery, composition and execution.

Most of these approaches support execution of composition schemes by providing execution engines, i.e., BPEL approaches use Oracle BPEL PM execution engine (Oracle 2005) or IBM BPWS4J (BPWS4J 2005), AI planners use OWL-S execution engines similar to the OWL-S API (Sirin 2004) provided by the University of Maryland.

Workflow-based approaches are static web services composition approaches involving manual intervention for discovery and composition of services. Semantic web based approaches achieve varying degree of automation in the process of composition (automatic discovery, semi-automatic composition).

4. Transparency

This criterion measures how transparent the process of composition (discovery, integration and execution) is from the composition participants.

For workflow-based approaches, end-user is transparent from the fact that the service presented to them in response to their request is a composed service, however the provider and composer has to work closely to integrate services in the workflow hence making the process opaque to them.

For AI planning based approaches, service requestor is transparent to the intelligent process of composition; however the process is semi-transparent to other participants. For example, the composer needs to be involved in the process of domain knowledge development and maintenance while tools assist them in converting semantic web services processes into planner domains. This knowledge is supplied to the planner in terms of operators and methods of services in order for planner to build composition plans. The service provider has to provide semantically enabled service but is transparent from the process of composition. Similarly, other semantic web based approaches offer complete transparency to end-users while requires some level of attention from service providers and composers.

5. Scalability of composition

Composing two services, however, is not the same as composing 10 or 100. In a real-world scenario, end users will typically want to interact with many services — consider the classic holiday booking scenario — while enterprise applications will invoke chains of possibly several hundred services Milanovic 2004).. Therefore, one of the critical issues is how the proposed approaches scale with the number of services involved. In BPEL, multiple service composition is somewhat tedious because XML files start to grow offering the approaches relying on BPEL as final composition scheme limited scalability (CSP based approach). OWL-S has similar issues and is propagated to the approaches that rely on using OWL-S process as final composition scheme (i.e., AI planning, software agent). Approaches that utilize bespoke XML schemas for final composition scheme (i.e., software synthesise approaches output synthesized XML schemas) also face similar challenges.

## 5. CONCLUSIONS

This contribution provides survey of two prominent categories of Web services composition approaches. The first approach, largely endorsed by the industry, borrows form business processes' workflow management theory to achieve the formalization necessary for describing the data flow and control in the composition scheme. The second approach, mainly promoted by the research community, aspires to achieve more dynamic composition by semantically describing the process model of Web service and thus making it comprehensible to reasoning engines or software agents.

The comparison made in this paper has shown that workflow based approaches are preferred by organizations as here-and-now and practical, albeit static, composition technique that robustly supports their business needs; while dynamic Web services composition approaches holds better futuristic potential that can serve a great range of business domains. In such kind of composition participating services can be external and public. User can specify parameters for the successful composition and the composition is performed at the run-time. The solution addresses the problems of identifying candidate services, composing them, and verifying closely that they satisfy the request.

At the end of this literature survey we concluded that despite the enthusiasm of the research community about the semantic web, there is still some way to go for creating a unifying framework facilitating the interoperation of intelligent agents or reasoning engines attempting to make sense of semantic Web services.

## REFERENCES


Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S. 2003. Business Process Execution Language for Web Services Version 1.1.

BPWS4J, "Business Process Execution Language for Web Services Java Run Time". http://www.alphaworks.ibm.com/tech/bpws4j.

Cardoso, J., Sheth, A. 2005. Adaptation and Workflow Management Systems. International Conference WWW/Internet. Lisbon, Portugal: 356-364.

Dean, M., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P. F., Stein, L. A. 2005. Semantic Markup for Web Services, OWL-S version 1.1.

Hammond, K. 1986. Learning to anticipate and avoid planning problems through the explanation of failures. Fifth Conference on Artificial Intelligence, AAAI86. Philadelphia, USA, Morgan Kaufmann. 1: 556-560.

High, R., Kinder, S., Graham, S. 2004. An Architectural Introduction and Overview. IBM's SOA Foundation

Kavantzas N et al, 2004. Web Services Choreography Description Language (WS-CDL) Version 1.0, http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/

Kuter, U., Sirin, E., Nau, D., Parsia, B., Hendler, J. 2005. "Information gathering during planning for web service composition." Journal of Web Semantics 3(2-3): 183–205.

Levesque, H., Reiter, R., Lespérance, Y., Lin, F., Scherl, R. 1997. "GOLOG: A logic programming language for dynamic domains." Journal of Logic Programming 31: 59-84.

Leymann F at al. 2002. "Web Services and business process management". *IBM Systems Journal*, Volume 41-2, 2002, 198-211.

Limthanmaphon, B., Zhang, Y. 2003. Web service composition with case-based reasoning. the Fourteenth Australasian database conference on Database technologies. K. Schewe, Zhou, X Adelaide, Australia, Australian Computer Society, Inc. Darlinghurst, Australia. 143: 201 – 208.

Madhusudan, T., Zhao, L. J., Marshall, B. 2004. "A case-based reasoning framework for workflow model management." Data & Knowledge Engineering archive 50(1): 87-115.

Martin D et al, 2004, "Bringing Semantics to Web Services: The OWL-S Approach", *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*.

Milanovic, N., Malek, M .2004. "Current solutions for Web service composition." IEEE Internet Computing 8(6): 51 - 59.

Oracle BPEL Process Manager (PM), http://www.oracle.com/technology/products/ias/bpel/index.html.

Osman T et al, 2005. An Integrative Framework for Traffic Telematics Web Services, to be appeared in the *Parallel and Distributed Computing Networks Conference(PDCN 2005).*

Peer, J. 2004. PDDL based Tool for Automatic Web Service Composition. In Principles and practice of semantic web reasoning, Springer Verlag, Berling, Germany: 15.

Porter, B. W., Bareiss, R. E. 1986. PROTOS: An experiment in knowledge acquisition for heuristic classification tasks. First International Meeting on Advnces in Learning (IMAL) Les Arcs, France: 159-174.

Russell, S., Norvig, P. 2003. Artificial Intelligence: A Modern approach, Prentice Hall.

Sirin, E., Parsia, B. 2004. Planning for Semantic Web Services. In Semantic Web Services Workshop at 3rd International Semantic Web Conference. Hiroshima, Japan, Springer Verlag, Berlin, Germany.

Thatte, S. 2001. XLANG Web Services for Business Process Design.

van der Aalst, W. M. P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P. (2003). "Workflow Patterns." Distributed and Parallel Databases 14(3): 45.

WSIF (2005). WSIF Apache Software Foundation Web Services project.

Wu, D., Parsia, B., Sirin, E., Hendler, J., Nau, D. (2003). Automating DAML-S web services composition using SHOP2. 2nd International Semantic Web Conference (ISWC2003).

Zisman, M. D. (1977). Representation, Specification and Automation of Office Procedures. Warton School of Business, University of Pennsylvania. PhD.

## AUTHOR BIOGRAPHIES

**DHAVALKUMAR THAKKER** took his Masters degree in Data Communication Systems at Brunel University, London-UK. He also received his bachelors degree in engineering from India. He is a PhD student of Nottingham Trent University, School of computing and Informatics. His current research interests are distributed computing technologies, Web Services composition, Semantic Web and Ontologies. His email address is dhavalkumar.thakker@students.ntu.ac.uk.

**TAHA OSMAN** is senior lecturer at the Nottingham Trent University, UK. He received a B.Sc honours degree in Computing from Donetsk Polytechnical Institute, Ukraine in 1992. He joined the Nottingham Trent University in 1993 where he received an MSc in Real-time Systems in 1994 and a PhD in 1998. His current research peruses the utilisation of semantic web technologies in web services composition and information retrieval. His email address is taha.Osman@ntu.ac.uk.

**DAVID AL-DABASS** is visiting professor at the Nottingham Trent University, UK. His research work explores algorithm and architecture for machine intelligence. His email address is david.al-dabass@ntu.ac.uk and web page is http://ducati.doc.ntu.ac.uk/uksim/dad/webpage.htm.