

Achieving Web Services Composition – a Survey

V. Portchelvi*, V. Prasanna Venkatesan, G. Shanmugasundaram

Dept. of Banking Technology, Pondicherry University, Pondicherry, 600 000, South India

Abstract Web Services composition development is too complex and dynamic. Software engineering approach is adopted to develop composite service systematically. Composition development process involves modelling, discovering & binding, publishing and invoking a composite service. This paper aims to make the composition development process clear by classifying the literature on web services composition into the phases of composition development process. As a result of this, the definition, requirements, approaches, techniques and evaluation of an end-to-end web services composition development process is made clear. The current research area which requires a lot of attention is identified and discussed at the end to further extend the topic.

Keywords Service Composition, Web Services Composition, SOA, Service Design, Composition Approaches

1. Introduction and Motivation

Automation and Integration of business processes requires tools and technologies that facilitate them. Service –Oriented Architecture (SOA) provides a cost-effective solution to this issue. SOA is emerging as the premier integration and architecture framework in today's complex and heterogeneous computing environment. Previous attempts didn't enable open interoperable solutions, but relied on proprietary APIs and required a high degree of coordination between groups. SOA can help organizations streamline processes so that they can do business more efficiently, and adapt to changing needs and competition, enabling software as a service concept. SOA is an architectural style for building software applications that use services available in a network such as the web. SOA is realized through a standards-based technology called web services. With the advent of web services technology this proprietary interfaces and data format are replaced with low cost ubiquitously supported standards for interfaces and data that work as well across the firewall as within it. The ability of web services to reach beyond the firewall, the loose coupling between applications encouraged by web services interfaces, and the wide support for core web services standards are the key reasons why web services technology promises to make integration of applications both within the enterprise and between different enterprises significantly easier and cheaper than before. To make the business process executable, the abstract view of the activities in the process must be instantiated with the web services that realize the

activity. In order to realize the automated Business-to-business interactions, web services composition is required wherein many services belonging to different enterprises are composed together and executed to realize complex business functionality. Web services composition must enable us to compose services automatically on- the- fly and on demand in order to realize business processes that can be created on demand. The benefit is virtual enterprises are created dynamically which shorten delivery times, increase product quality, deliver personalized services, decrease transaction costs, and accommodate short-term cooperating relationships[5]. In this paper we aim to survey the complete research done in web services composition and to provide an overview of the service composition approaches. Our aim in this survey work is to focus on a Web services composition, aiming to arrange multiple services in a meaningful manner, to supply complex needs.

The complete Literature on the services composition process can be classified as

- papers that discuss about the background, motivation and models to develop Service composition[1][2][5][12]
- papers that discuss about the composition approaches developed based on the requirements of a particular situation (requirements based)[1][7][15][27][30][34][35]
- papers that discuss about the solution for end-to-end composition (Service composition life-cycle phases based) [3][9][10]
- papers that discuss about the composition approaches & techniques[4][6][28][29][30][31][32][33][34][35][36]
- papers that evaluate the service composition approaches (comparison or evaluation based) based on certain common evaluation

criteria[1][2][3][9][11][13][14][15][16][17][18][20][23][24]

Based on the above classification, this paper is organized as follows. Section 2 presents the definition of Web service

* Corresponding author:

portchelvi_vasu@yahoo.co.in (V. Portchelvi)

Published online at <http://journal.sapub.org/se>

Copyright © 2012 Scientific & Academic Publishing. All Rights Reserved

composition. Section 3 is to state the requirements of Web service composition. Section 4 provides an overview of composition approaches. Section 5 is to describe the whole composition process. Section 6 discusses about the evaluation of existing approaches. The last section concludes the paper.

2. Web Services Composition

In the literature the terms such as “Web services composition”, “web services flow”, orchestration and choreography were used to describe the composition of the web services in a process flow[5]. All these provide a means to weave web services into meaningful business processes.

2.1. Composition Models

In a WS environment, a workflow represents a composition of Web Services (WS). When each activity of a workflow is implemented by a component WS, we obtain a Composite Web Services (CWS). Several CWSs can be associated to the same workflow, depending on the assigned component WSs. The orchestration of the component WSs is defined by specifying dependencies between them. These dependencies are defined by the associated workflow patterns and by the transactional properties. The first ones specify how WSs are coupled and how the behaviour of certain WSs influences the behaviour of other one(s), while the transactional properties specify the behaviour of certain WSs in case of failure. The process of producing a web service composition can be generally viewed as starting with some global properties or dependencies (i.e. composition flow) that must be specified. Given the global dependencies the individual web services has to be orchestrated so that they collectively satisfy the global dependencies. Depending on whether and when the knowledge of global dependencies is known there are two approaches to compose web services.

i) Mediated Approach (Orchestration Model)

In this approach all global dependencies are known to at least one service (called the mediator) prior to the execution.

ii) Peer- to – Peer Approach (Choreography Model)

In this approach every individual service knows only a subset of but not the full global dependencies.

2.2 Composition Engine

Since a business process involves complex, related, and structured activities, it requires a stateful environment for invoking a chain of web services that implement the business process. Services representing core business logic might participate in complex processes and orchestrated by a process-centric service (orchestration engine). Process-centric services do not usually have application state but they have state related to the process itself. The process state includes information regarding

i) Process participants (people & services)

ii) Input from participants

iii) The actual position within the process flow

iv) Basic rules

Process-centric services are highly dependent on other services. They provide the glue to bind the core business services. Process-centric services must co-ordinate complex activities that can span more than one person, several major business entities, multiple locations or long periods of time. Seamless composition of web services has enormous potential in streamlining B2B or EAI. Orchestration refers to an executable business process that can interact with both internal and external web services. The interactions occur at the message level. They include business logic and task execution order and they can span application and organizations to define a long-lived, transactional, multistep process model. Orchestration always represents control from one party’s perspective.

2.3. Composition Standards

Initially Web services technology largely focused on interoperability of heterogeneous systems based on the messaging foundation supported by WSDL and SOAP. But in order to fully automate the business process, this first generation web service technology is not sufficient. The complete automation of web services requires the ability to specify work flow, security requirements, transaction management and other critical information related to business process context. Such information is generally specified in a business process model. Web services composition that makes the process executable has to take care of the above mentioned information. So a standards based approach is required for composing web services to create a high level business process.

In the literature it has been stated that current Web services Composition does not have a reference model. It adopts the general web services stack as a defacto standard for web services interactions, and adopts standards such as BPEL4WS and WSCI[2] for connecting these web services together to form more meaningful business processes.

3. Web Services Composition Domain Requirements

This section is aimed to group the core technical requirements of WSC domain that are spread across the literature. The requirements are the extension of the basic requirements which have been discussed mostly in the literature. The web service environment is highly dynamic in nature. In order to perform services composition in this domain, the composition approaches developed must meet the following requirements of WS domain.

• Services domain complexity (large state spaces)

Exploratory

The number of services available over the web increases drastically during the recent years and it is expected to have a huge web service repository[1].

Volatile

Web services are created and updated on the fly. The environment is characterized as volatile[1] which means

services enter into and exit from the domain dynamically. That is the environment keeps changing. The changes relate also to the QoS attributes, whose values might evolve in time.

Uncertain / Non-Deterministic

Distributed computing environments are inherently uncertain: invocation of remote WSs may potentially produce unexpected responses[7][27][34]. A response could depend on whether the WS is operating correctly and on specific external real-world situations.

Partially observable

The services in this environment are only partially observable i.e. their internal status and variables[7][27] are not accessible.

Heterogeneity

Web services can be developed by different organizations, which use different concept models to describe the services. There does not exist a unique language to define and evaluate the web services in an identical means.

• Composition Performance (Fast & Scalable Composition)

The composition process is a high computation demanding process. The computations required to compose a service satisfying user goal might be significantly time consuming. This is due to the reason that the number of services available over the web increases drastically during the recent years and it is expected to have a huge web service repository to be searched[30].

• Adaptability

Web services are created and updated on the fly, thus the composition system needs to detect the updating at run-time and a decision should be made based on the up to date information.

• Failure Resilience

A key idea of SOA is that service requestors and service providers can be and usually are from different organizations which are under certain circumstances residing in different parts of the world. The probability of errors such as communication failures and changes of services in distributed environments is higher than in mono-organizational environment. If an exception[35] occurs it would be possible to re-compose a service composition with the aim that the new composition with probably different services works without fail.

• Composition Optimization

The ability of service composition approach to optimize the composition using non-functional parameters is called optimization. Although satisfying the functional requirements is important for building the WSC, optimization of non-functional preferences may be equally crucial. For example, a WSC that minimizes the response times and WS costs, and guarantees a basic level of availability etc. is required[34].

• Composition Correctness

The composition approach must be able to guarantee the correctness of the composition[15]. The developed composition behaviour must conform to its requirements.

• Composition Execution

The composition execution can be done in centralized or distributed manner.

4. End-to End Composition Process

The service composition life-cycle involves the following phases to develop an end-to-end service composition. The purpose of each phase in the life-cycle has been detailed below:

Table 1. Service Composition Life-Cycle Phases & its purpose

S.No.	Phase	Purpose
1.	Specification	User specifies initial state, goal including QoS constraints
2.	Planning	Algorithms/techniques are used to automatically generate a workflow plan in which the component services are ordered and connected.
3.	Validation	To check the composition correctness
4.	Discovery	To find the actual component services
5.	Execution and Monitoring	To execute the composite service and to monitor for exceptions and failures

In[9], the authors state that most of the research surveys on service composition tend to either focus on one particular emerging technology (such as workflow-based, AI planning-based, ontology-based, etc) or be domain-specific. Service composition life-cycle phases such as service discovery, process model creation, Process model verification and Execution are detailed in this paper. In the literature, only little efforts had been put to develop end-to-end composition approaches. In[10], the survey states that recent literature on web composition provides too narrow a scope. Some only discuss a portion of the WSCE process in detail while others provide an overall view, but with a narrow perspective. So,[10] proposes a high-level web service composition framework that addresses the composition process from start to finish. In[3], the various current solutions available for end-to-end service composition are discussed. To realize the Web Services Composition and Execution (WSCE) process, various solutions are available and they are discussed in this paper.

5. Composition Approaches

This section is to describe about the methods, techniques and tools that are used to develop a service composition. Our focus is to bring out the current research efforts involved to develop a composition as per the requirements of services domain. The literature shows that the work in developing appropriate composition approaches is being done in the industry side as well as in the academic research side[12].

5.1. Industry Approach

Since there is no standards based approach for service composition, composition of services in the industry is done by adopting the general web services stack as a defacto

standard for web services interactions, and adopts standards such as BPEL4WS and WSCI for connecting these web services together to form more meaningful business processes. BPEL4WS is a specification that models the behavior of web services in a business process interaction. The specification provides an XML-based grammar for describing the control logic required to coordinate web services participating in a process flow. This grammar can then be interpreted and executed by an orchestration engine, which is controlled by one of the participating parties. The engine coordinates the various activities in the process, and compensates the system when errors occur. This type of specifying composition is too low level (programming level) and primitive. These approaches use the traditional programming language techniques to perform composition process. They use specific language structures and control patterns for composing services and perform service composition at the micro level. This is purely a static approach and there is no way for adaptation of the process and thus not allowing it to change at runtime. This leads composition to be done statically and manually.

In this approach, the composition can be viewed as a function-centric composition based on the reason that for composing services the reasoning behind connecting two component services depends only on the service input and output i.e. only the syntactic description of service. These types of approaches come under syntactic based services composition.

In Syntactic-based approach currently there are two main approaches for WSC i.e. Orchestration and Choreography. Orchestration combines available services by adding a coordinator (the orchestrator) that is responsible for invoking and combining the single sub activities (BPEL4WS) and Choreography instead does not assume a coordinator but rather defines complex tasks via the definition of the conversation that should be undertaken by each participant; the overall activity is then achieved as the composition of peer-to-peer interactions among the collaborating services.

5.2. Academic Approach

But in the real world services environment, services are volatile, exploratory and dynamic. So, composition approaches exposing higher degree of dynamism are required. Since the service environment is a highly flexible and dynamic environment in which new services become available on a daily basis and the number of service providers is constantly growing, performing service composition manually is an error-prone, tedious process. Static specification of service composition will not be able to cope up with the dynamics of the services environment. So, the specification of services composition process must be generated at run-time based on the request and this should be automated. This type of composition approach is called dynamic services composition and it overcomes the two limitations of static approach namely, manual and static specification of service composition process. For this

purpose, the academic research community utilizes semantic web perspective to solve the composition automatically and dynamically. In this approach, the composition can be viewed as a process-centric composition based on the reason that for composing services the reasoning behind connecting two component services depends on the service input, output, pre-conditions and effects i.e. the semantic description of service. The semantic descriptions provide software agents with a way to automatically reason about the service's semantics, i.e. the preconditions and effects. In an environment of semantically annotated services, users who need to achieve certain goals could be assisted by software agents who automatically identify and, if necessary, dynamically compose services in order to accomplish the user's goals, which may be either explicitly stated or derived from the situation the user is in. These approaches are grouped under semantic based services composition.

Current WS technologies address only the syntactic aspects of services and thus provide a set of rigid WSs that cannot adapt to a changing environment without human intervention. Currently there are two principal models available for semantic based WSC i.e. OWL-S and WSMO. OWL-S is an effort to define ontology for the semantic markup of WSs, intended to enable the automation of service discovery, invocation, composition, interoperation and execution monitoring by providing appropriate semantic descriptions of WSs. The WS Modelling Ontology WSMO is an effort to create ontology to describe various aspects related to semantic web services, aiming to solve the integration problem. The goal of both initiatives is to provide a standard for the semantic description of WSs.

6. Composition Techniques

In the literature, most of the dynamic service composition approaches concentrate much on the automated generation of on-demand service composition process specification. In order to achieve automated on-demand service composition, researchers related this to the automated problem solving techniques available in AI. Different branches of Artificial Intelligence (AI) research develop frameworks for describing problems and resources along with technologies for automated problem resolution and resource usage. These frameworks support both client side as well as service side. On the service side, it helps in defining the computational resources such that they can be allowed in automated usage. On the client side, it helps in describing the problem to be solved and the resolution process. The client side should allow specifying the objectives to be solved from the client's perspective without regard to their technical resolution, thereby providing support for the client-side that is decoupled from technical service usage requests. The client side description should include all information required for automated problem resolution by intelligent mechanisms for automated resource detection, combination and usage. An architecture that supports this type of client side descriptions

is called goal-driven architectures. In this architecture, the user needs to specify only his wish or the problem to be solved while the intelligent mechanisms handle the resolution process automatically. The goal-driven architecture mainly focuses on the client side rather than the service side. The reason is, only the client side provides necessary required information to the intelligent mechanisms for automated problem solving.

6.1. AI Planning Technique

In AI there are lot of approaches and technologies available which supports the above mentioned automated goal-driven problem solving based on the client side problem specification. One such sub discipline of AI, AI planning techniques[22] have been explored and exploited for developing automated on-demand services composition. Most of the approaches for automated on-demand services composition discussed in the literature are based on the AI Planning techniques. In AI planning, on the client side, the user specifies his objectives to be achieved and the intelligent planning algorithm takes the objective of the client as goal and produces a plan for the goal resolution process. A plan contains a sequence of actions that will lead to the goal. AI Planning tries to construct the entire plan at the beginning. Planning algorithms can be classified into classical planning algorithms and non-classical planning algorithms. The former is suitable for environments that are fully observable, deterministic, finite, static and discrete whereas the latter is suitable for partially observable or stochastic environments.

Table 2. Approaches and their supporting features

S.No.	Approach	Feature supported
1.	Zeng et al.[33] Based on rule inference engine	Services are annotated with QoS features.
2.	Pistore et al.[28] Based on model-checking planner MBP	Non-deterministic services and partial observability of service effects.
3.	Sirin et al.[29] Based on Hierarchical Task Network (HTN) Planning	Hierarchical decomposition of complex task to atomic-invokable-services.
4.	Berardi et al.[31] Based on automatic synthesis of finite state machines	Non-deterministic finite state machines are used to support real-world , external services.

Since the services environment is characterized by dynamic, open world settings, non-deterministic, uncertain, partially observable etc; classical planning algorithms cannot be applied. Therefore classical planning algorithms are not directly applicable for the automated services composition process. In the literature, researchers explored the

requirements for automated on-demand services composition and based on that they extended the planning algorithms (non-classical planning algorithms) that are well suited for the services domain.

Existing non-classical planning based composition approaches in the literature for automated service composition differ by the functionality they provide. Each approach considers its own set of features that are necessary to model the real-world services and composition. Few well-known approaches had been listed in the table below:

Some part of the literature survey for services composition domain shows that approaches are developed for automatic plan generation. They concentrate only on the planning phase of service-composition life-cycle because the services composition domain needs plans to be generated with the following characteristics and the non-classical planning techniques must produce them. In the services composition domain plans that are to be generated automatically by the planners must have the following characteristics[12][25]

- Plans need complex control structures with loops, non-determinism and conditionals i.e. plans are thus expressive enough for representing the flow of interactions of the synthesized composed service with the other services and expressive enough for representing the required observation over other .

- Plans can produce new objects at execution time and the objects managed may have a complex structures and descriptions.

- Use of non-functional attributes such as cost or quality which facilitates the choice of the plan.

- Plans need to support semantic constructions such as hierarchies (abstractions) as well as compatibility with the semantic service descriptions standards such as OWL-S & WSMO.

- Plans must also consider the definition of extended goals involving complex conditions on process behavior.

Most of the AI planning based automated service composition approaches in the literature concentrates much on the automatic generation of plans with the above characteristics.

In the other part of the literature, various frameworks are available for automated service composition. These frameworks predominantly utilize AI planning techniques to achieve automated service composition. Automated service composition frameworks (AI Planning based) are compared based on the following features:

From the table we can conclude that the above mentioned Service composition frameworks have not been designed for a common purpose. Each one is designed for one feature. The common objective of the above mentioned frameworks is automation. Apart from that they cannot be compared based on any common feature. The composition approaches used so far is chosen based on the concrete situation. It is unable to put them into a position of competitors.

Table 3. Comparison of composition Approaches

	SHOP2[29]	ConGolog	SWORD[4]	ASTRO[27]	TL PLAN[35]	Haley[34]
Extended Goals	P	Y	N	Y	P	N
Complex actions	Y	Y	N	Y	P	Y
Automatic service discovery	N	P	P	P	Y	P
On-the-fly recomposition	N	N	N	N	Y	Y
Monitoring for Non-determinism	P	P	N	Y	P	Y
Composition failure recovery	N	N	N	N	Y	Y
Execution failure recovery	R	N	R	R	PR & R	-
QoS based modeling	N	N	N	N	Y Scalability Performance	Y Cost
Service markup	OWL-S	OWL-S	XML	BPEL 4WS	OWL-S	SAWSDL

Y- Feature present in the Framework

N- Feature not present

P-Feature partially present

PR- Proactive

R - Reactive

6.1.1. Advantages of Planning Techniques

AI techniques have been used to support different key aspects of the management of service compositions, including tasks such as their generation, allocation of resources, execution, monitoring and repair[26]. For instance,

- knowledge representation techniques have been exploited to provide suitable semantic annotations of services
- planning has been applied to an automatic generation of the workflows composing the services
- scheduling has been applied to resource allocation and workflow optimization
- agent techniques have been applied to support a dynamic adaptation of the workflows

6.1.2. Drawbacks of Planning Techniques

There are lots many issues remain to be resolved in using AI planning techniques for Web Services composition. HTN, Golog, classic AI planning, rule-based planning, model checking, theorem proving, etc. are the AI Planning techniques used in current approaches for automated web services composition. These approaches assume that relevant service descriptions are initially loaded into the reasoning engine and that no service discovery is performed during composition. In fact, these approaches do not fit an environment where a large number of dynamically changing service advertisements are published over Internet. But overcoming both discovery and composition of services is the key to automatic generation of executable process[32]. In order to support dynamic and automated tasks such as discovery, selection and composition, semantic model of web services (Semantic Web services) are used. The semantic feature of Web services allows not only a high level of semantic description but also different kinds of inference

in order to ease automatic discovery, selection, composition and reasoning. In Semantic-based approaches, ontologies are used as data models throughout these types of approaches, meaning that all resource descriptions and all data interchanged during service usage are based on ontologies [37]. However, despite the merits and the importance of semantic information contained by services, some drawbacks existing in most semantic-based composition methods have prevented them moving forward. One drawback is the low efficiency brought by the direct reasoning algorithm. The other drawback is the way ontologies are used, since ontologies are currently a major technology for supporting service description and composition. Different organisations define ontologies in different ways, which may generate major problems of interoperability. Some approaches define manual mappings to deal with the ontologies interoperability problem; however these mappings or mediation techniques may be error prone and difficult to apply in realistic applications [38].

AI planning based approaches generates the composition specification completely and then executes it. This might cause failures in composition execution because of changes in the world that timely occur between the planning and its execution. Execution monitors have been applied to support a dynamic adaptation of composition. Having done a detailed survey, it is identified that one common area where most of the AI planning based frameworks lack their attention is this failure handling[35].

Hence Dynamic Composition process requires flexibility and autonomy which is not fully achieved using AI planning techniques.

6.2. Agent Technology

AI planning is the main stream method for automatic semantic web service composition (SWSC) research.

However, planning based SWSC method can only return service composition upon user requirement description and lacks flexibility and autonomy to deal with environment change. Deliberate agent architecture, such as BDI agent[36], is hopeful to make SWSC more intelligent. Similarly, the application of learning techniques (e.g. based on feedback from earlier runs) may be considered to improve the agent's intelligence.

7. Composition Approaches Evaluation

A significant number of evaluation papers have been published in literature for comparing web service composition approaches based on some criteria. The survey of [1] extensively evaluates the various available web service composition approaches based on classifications and compositional issues. In [2][12], the paper focuses more on web service composition languages like WS-BPEL with WSDL, OWL-S with Golog/Planning.[3] discusses Web Service Composition and Execution (WSCE). The frameworks are categorized as interleaved, Monolithic, Staged, and Template-based service composition and execution. In [9], the authors evaluate the web service composition frameworks based on the concept of service composition life-cycle. In [11], the authors evaluate the web service composition frameworks based on the level of automation of the service composition process.[13] is an evaluation paper specifically on dynamic service composition approaches. It summarizes the currently available dynamic web service composition approaches, evaluates the approach according to the framework and then gives an outlook to essential future research works.

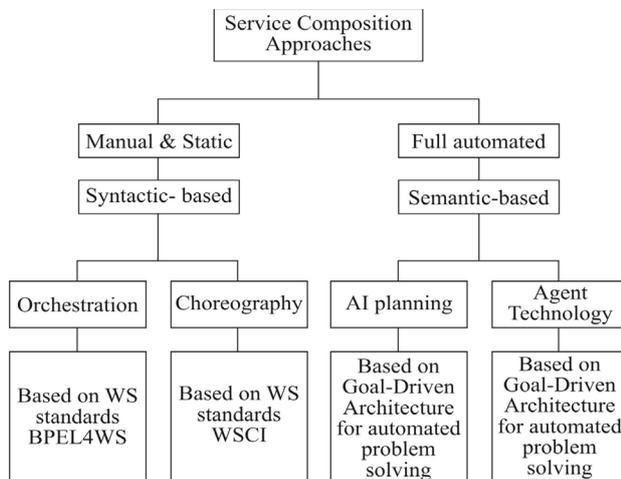


Figure 1. Service Composition Approaches and Techniques

8. Conclusions

The motivation behind this survey is to study the various researches carried out in the WSC domain. From the study it is found that the existing literature surveys on WSC lack comprehensiveness and therefore this paper puts an effort in

presenting a comprehensive survey on Web Services Composition. This has been done by classifying the various literatures in the service composition domain into five classes based on various composition development aspects and this classification enabled us to understand the evolution of the Web Services Composition approaches. Importantly the comprehensive study helped in finding research gaps that leads to opportunities.

Our future research work is to address the limitations of AI Planning techniques in WSC that requires lot of attention in Web Services Composition and has been discussed below. In the literature the focus is specific towards automating the composition process using AI Planning techniques. These works considered that the automation of process model generation is bottleneck for dynamic services composition and tried to solve it by AI planning techniques and has gained some maturity in it. But achieving Web Services Composition dynamically requires lot of flexibility and autonomy which AI Planning technique couldn't provide. Hence it is proposed to explore the other techniques to provide a more flexible, autonomous, dynamic and intelligent solution to web services composition problem.

REFERENCES

- [1] S. Dustdar, W. Schreiner, "A survey on web services composition", *Int. Journal of Web and Grid Services*, vol. 1, no.1, pp. 1-30, 2005.
- [2] Antonio Bucchiarone, Stefania Gnesi, "A Survey on Services Composition Languages and Models", in *Proceedings of Int. Workshop on Web Services Modeling and Testing*, pp. 51-63, 2006.
- [3] Vikas Agarwal, Girish Chafle, Sumit Mittal, Biprav Srivastava, "Understanding Approaches for Web Service Composition and Execution", in *Proceedings of First Bangalore Annual Compute conference*, Article no. 1, 2008.
- [4] S. Ponnekanti, A. Fox, "SWORD: A Developer Toolkit for Web Service Composition", in *Proceedings of the 11th International World Wide Web Conference*, 2002.
- [5] Chris Peltz, "Web services orchestration a review of emerging technologies, tools, and standards", Hewlett Packard, Co., January 2003.
- [6] B. Srivastava, "Automatic Web Services Composition Using Planning", in *Proceedings of Knowledge Based Computer Systems*, pp. 467-477, 2002.
- [7] P. Traverso, M. Pistore, "Automated Composition of Semantic Web Services into Executable Processes", in *Proceedings of 3rd International Semantic Web Conference*, pp. 380-394, 2004.
- [8] Soon Ae Chun, V. Atluri, and N. R. Adam, "Policy-based Web Service Composition", in *Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications*, pp.85-92, 2004.

- [9] Rodrigo Mantovaneli Pessoa, Eduardo Silva, Marten van Sinderen, Dick A. C. Quartel, Luís Ferreira Pires, “Enterprise Interoperability with SOA: a Survey of Service Composition Approaches”, in Workshop on Enterprise Interoperability, pp. 32-45, 2008.
- [10] Anya Kim, Myong Kang, Elias Ioup, Catherine Meadows, and John Sample, “A Framework for Automatic Web Service Composition”, sponsored by the U.S. Naval Research Laboratory’s Base Program, Program Element No.0602435N, April 2009.
- [11] Rao. J. and X. Su, “A Survey of Automated Web Service Composition Methods”, in Proceedings of the First International Workshop on Semantic Web Services Web Process Composition, 2004.
- [12] Srivastava. B, Koehler. J., “Web Service Composition - Current Solutions and Open Problems”, in Proceedings of ICAPS 2003 Workshop on Planning for Web Services, 2003.
- [13] Ravi Khadka and Brahmananda Sapkota, “An Evaluation of Dynamic Web Service Composition Approaches? ”, In: 4th International Workshop on Architectures, Concepts and Technologies for Service Oriented Computing, 2010.
- [14] Sayed Gholam Hassan Tabatabaei, Wan Mohd Nasir Wan Kadir, Suhaimi Ibrahim, “A Comparative Evaluation of State-of-the-Art Approaches for Web Service Composition”, in Proceedings of Third International Conference on Software Engineering Advances, pp. 488-493, 2008.
- [15] Maurice H. ter Beek, Antonio Bucchiarone, Stefania Gnesi, “Formal Methods for Service Composition”, Annals of Mathematics, Computing & Teleinformatics, vol. 1, no. 5, pp. 1-10, 2007.
- [16] B. Benatallah et al., Service Composition: Concepts, Techniques, Tools and Trends in Service-Oriented Software System Engineering-Challenges and Practices, pp. 48–66, Idea Group Publishing, USA , 2005.
- [17] J. Mendling and M. Müller, “A Comparison of BPML and BPEL4WS,” in Proceedings of 1st Conference Berliner XML-Tage, pp.305–316, 2003.
- [18] P. Wohed et al., “Pattern-Based Analysis of BPEL4WS”, Queensland University of Technology, Brisbane, Technical Report FIT-TR-2002-04, 2002.
- [19] M. P. Papazoglou et al., “Service-Oriented Computing: State of the Art and Research Challenges”, Computer, vol. 40, no. 11, pp. 38–45, 2007.
- [20] Juan Carlos Zuñiga, Jos’e J. P’erez-Alc’azar, Luciano Digiampietri, “Implementation Issues for Automatic Composition of Web Services”, Workshops on Database and Expert Systems Applications, pp. 201-205, 2010.
- [21] Benatallah, B., Dumas, M., Fauvet, M.C., Rabhi, “Towards patterns of web services composition”, In: Patterns and skeletons for parallel and distributed computing, London, Springer-Verlag, 2003.
- [22] Peer, J., “Web service composition as AI planning - a survey”, Univ. of St. Gallen, Switzerland, Technical report, 2005.
- [23] Ulrich KÄuster, Mirco Stern, Birgitta KÄonig-Ries, “A Classification of Issues and Approaches in Automatic Service Composition”.
- [24] Nikola Milanovic, Miroslaw Malek, “Search Strategies for Automatic Web Service Composition”, International Journal of Web Services Research, Vol.X, No.X, 200X.
- [25] Uciano A. Digiampietri, Jos’e J. P’erez-Alc’azar , Claudia Bauzer Medeiros, “AI Planning in Web Services Composition: a review of current approaches and a new solution”,
- [26] Marco Pistore , Jose Luis Ambite, Jim Blythe, Jana Koehler, Sheila McIlraith, Biplav Srivastava, “AI for Service Composition”, in workshop , 2006
- [27] M. Trainotti, M.Pistore, G. Calabrese F. Baron, G.Zacco P. Bertoi, G.Lucchese, P. Traverso, “ASTRO: Supporting Composition and Execution of Web services”,
- [28] Piergiorgio Bertoli, Alessandro Cimatti, Marco Pistore and Paolo Traverso, "A Framework for Planning with Extended Goals under Partial Observability", in Proceedings of ICAPS-03, 2003.
- [29] Wu, D., Sirin, E., Hendler, J., Nau, D., and Parsia, B., “Automatic Web Services Composition Using SHOP2”, in Twelfth World Wide Web Conference, 2003.
- [30] Peter Bartalos, “Effective automatic dynamic semantic web service composition”, ACM, 2010
- [31] D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, and M. Mecella, “Automatic composition of transition-based semantic web services with messaging”, in Proceedings of VLDB, 2005.
- [32] Sayed Gholam Hassan Tabatabaei, Ph.D. candidate, Universiti Teknologi Malaysia, “Integrating Discovery and Composition of Semantic Web Services Based on Description Logic”, Journal of Computer Science , Informatics & Electrical Engineering, Volume 3, Issue 1, 2009.
- [33] Huiyuan Zheng, Jian Yang, Weiliang Zhao, QoS, “Analysis and Service Selection for Composite Services”, in the Proceedings of International Conference on Services Computing, 2010.
- [34] Haibo Zhao Prashant Doshi, “A hierarchical framework for logical composition of web services”, Published online: Springer-Verlag London Limited, 2009.
- [35] Maja Vukovi’c, IBM T.J.Watson Research, Evangelos Kotsovinos, Peter Robinson, “Application development powered by rapid, on-demand service composition”.
- [36] LI Huan, QIN Zheng, YU Fan, Qin Jun, YANG Bo, “Automatic Semantic Web Service Composition via Agent Intention Execution in AgentSpeak”, International Conference on Web Intelligence, IEEE/WIC/ACM, 2007.
- [37] Freddy L’ecu’e1,2, Alain L’eger Ecole Nationale Sup’erieure, “Semantic Web Service Composition Through a Matchmaking of Domain”, in Proceedings of the European Conference on Web Services , IEEE, 2006.
- [38] Kaijun Ren, Jinjun Chen, Nong Xiao, Weimin Zhang , Junqiang Song, “A QSQL-based Collaboration Framework to Support Automatic Service Composition and Workflow Execution”, in the 3rd International Conference on Grid and Pervasive Computing – Workshops, IEEE, 2008.