ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 2, Issue 12, December 2014

# A Comparative Study on Web Life Cycle Activities & Composite Web Services

Atul Garg [1], Jyoti Parashar [2]

Associate Professor, MMICT&BM (MCA), Maharishi Markandeshwar University, Mullana, Haryana, India[1]

Research Scholar, MMICT&BM (MCA), Maharishi Markandeshwar University, Mullana, Haryana, India[2]

**ABSTRACT:** Web is the main source of communication, E-commerce, online shopping, research etc. Web Service life cycle is main fundamental to use Web Service. The use of Web is changed its purpose and method from its beginning to this era. And, as the use of Web is changed its life cycle is affected. In this paper various Web Service Life Cycles and their approaches for their need in this era are analysed and compared. The paper also includes the work flow of Web Services including the validation and security issues of Web Services.

**KEYWORDS**: Web Services, Web Service Life Cycle, Composite Services.

## I. Introduction

Internet is becoming the prominent paradigm for distributed computing and electronic business. Web Services refers to independent, Web applications accomplished not only of performing business activities on their own, but also possessing the ability to engage other Web Services in order to complete high-order business transactions. Such as, Web Services comprise online payments, stock market related services etc. The platform neutral nature of Web Services creates the opportunity for developing composite services by using existing atomic or composite services possibly offered by different organizations.

By the author [34], a Web Service is a piece of software applications whose interface and binding can be clearly defined, describe and discover as XML artifacts. It supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols. Standards are key enablers of Web Services [36]. Major industry players took a lead to set up crucial standards. Really, it facilitated the adoption and deployment of Web Services [35]. As the Web Services paradigm becomes popular more applications are created and deployed as Web Services, the need for developing new solutions tackling the composition of Web Services becomes manifest. However, emerging Web Service standards (e.g., WSDL, UDDI, WSFL and BPFL4WS) and existing methods are not sufficient for realizing the goal of flexible and dynamic composition of Web Services. Although, some preliminary work has been covered in the area of Web Service composition based on aspects of workflow-like service integration, service conversation and B2B protocol definition [1, 3 & 26].

### A. Challenges of Web Service Composition:

The real challenge in services composition lies in how to provide a complete solution that supports the entire life cycle of services composition. The main three phases are planning, definition and implementation [44]. Planning phase is related to the user's requests. Planning is a composition plan that how to complete the user's tasks which maps his/her requests needs to be proposed firstly and then the candidate atomic or composite services that possibly complete these tasks need to be discovered correspondingly. During this phase, every task from the user side needs to be mapped to each service. The outcome of this phase is the combination of a composite service out of desirable, or potentially available, atomic services and the structure of a composite service is generated and formed. At the definition phase, the internal dependencies of the composite service need to be clearly defined and specified. Thus the first two phases within the entire life cycle of services composition are very important and they correspondingly have two interesting research points, one is how to systematically plan and model the structure of a composite service and another is how to clearly specify the inter-relationships of a composite service. If these two issues can be properly solved, the development and implementation of Web Services composition would be greatly facilitated. However, the existing

10.15680/ijircce.2014.0212016

standards and approaches haven't systematically addressed these issues or are not enough to effectively tackle these issues.

**B. Web Services**

The Web has become the means for organizations to deliver goods to provide online services and for customers to search and retrieve services according to their needs. Web Services reduces the cost of e-business, to deploy solutions faster and to open up new opportunities. Web Services are self-contained; Internet enabled applications capable not only of performing own business activities, but also possessing the ability to engage other Web Services in order to complete higher-order business transactions. Easy Web Services provide simple functions such as credit checking, authorization, inventory status, weather reports etc., while composite services may appropriately unify disparate business functionality to provide a whole range of automated processes such as insurance brokering, travel is planning, insurance liability services or package tracking. Several software vendors and consortium are providing platforms (such as IBM's Web Sphere, Microsoft's .NET etc.), languages and description models for service representation and discovery such as Web Service Description Language (WSDL) and Universal Description Directory and Integration (UDDI), which offer uniform representation and access to Web Services. At the same time, for business marketers, Web Services improving inter-organizational relationships and generating new revenue streams [5]. Further, Web Services can be considered a further development of e-commerce or e-business, because they are service-focused business paradigms that use two-way dialogues to build customized service assistance, based on knowledge and experience about users to build strong customer relationships [6].

Web Service lifecycle is a fundamental topic for Web Services and Service-Oriented Computing (SOC). Web Service lifecycle is the basis for engineering and managing activities in Web Services. For example, many techniques, approaches, methods have been proposed to facilitate or support the main stages of the entire Web Service lifecycle [7]. Many Web Service lifecycles have also been proposed to improve Web Services with their applications. If the main parties and their demands are ignored in Web Services, the strong development of Web Services might be problematic, because ignorance of demands in economy and business will lead to economic crisis, like the current global financial crisis. Web Service requesters denote Web Service users, buyers, customers, consumers, receivers and their intelligent agents. Numbers of Web Service users are increasing day by day from its evolution to this era (as shown in Fig. 1).

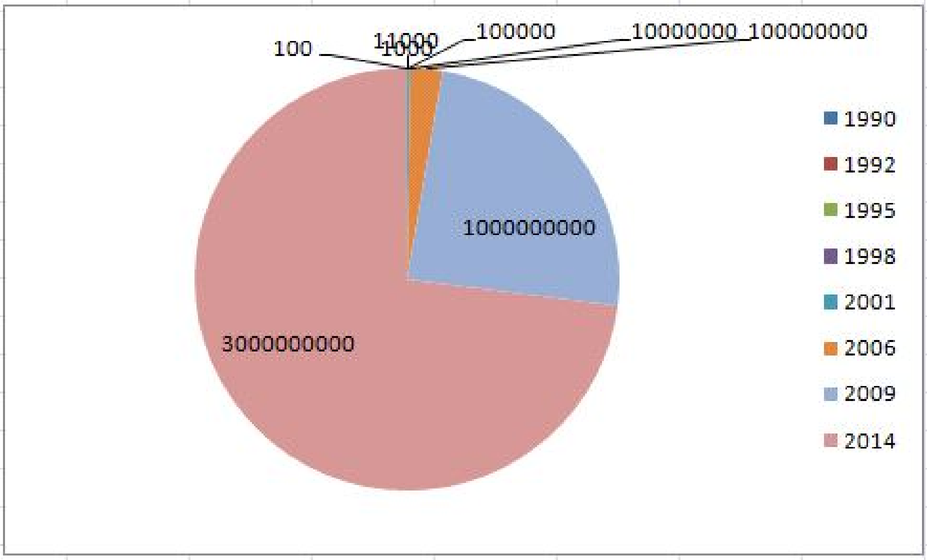

**Fig 1: Web Service Evolution**

The authors in [4 & 44] concentrates on first two interesting issues and propose the well-defined notions "Composition Structures" to plan and model the structure of a composite service during the planning stage of services

10.15680/ijircce.2014.0212016

composition and a specification mechanism that effectively tackles the definition stage of the entire life cycle of services composition, as it can clearly specify the internal dependencies of the composite service which effectively facilitated the implementation of services composition which is defined by Xiang in [44]. In the planning stage of services composition, our objective is to easily generate the composition process and to support the composer in selecting most suitable services [4].

Although Web Services composition has been extensively studied in the past decade, techniques are still not fully mature yet with several open issues remaining. Moreover, the rapid rise and adoption of new computing paradigms such as cloud computing, social computing and Web in recent years also presents compound challenges in this area [35]. The authors [35] identify several directions for future research on services composition.

In section II, research work done by the related research is discussed. In section III Stages of Web Service Life Cycle are analyzed and compared. In last section of this paper conclusion of the study is discussed.

## II. Related Work

Many researchers are working on the same work. Most of the work in service composition has focused on using workflows either as engine for distributed activity co-ordination or as a tool to model and define service composition. The authors [18] discuss the development of a platform specifying and enacting composite services in the context of a workflow engine. The e-Flow system provides a number of features that support service specification and management, including a simple composition language, events and exception handling.

A Web Service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols. Standards are key enablers of Web Services [21 & 24]. Major industry players took a lead to set up crucial standards. The researchers of [25, 26 & 27] focused on three key XML-based standards: Simple Object Access Protocol (SOAP) [25], Web Service Description Language (WSDL) [26] and Universal Description, Discovery and Integration (UDDI) [27]. SOAP defines a communication protocol for Web Services. WSDL enables service providers to describe their applications. UDDI offers a registry service that allows advertisement and discovery of Web Services.

The Web Services paradigm promises to enable rich, flexible and dynamic interoperation of highly distributed and heterogeneous Web-hosted Services. Substantial progress has already been made towards this goal, e.g., emerging standards such as SOAP, WSDL, Business Process Execution Language (BPEL) and industrial technology (e.g., IBM's Web Sphere Toolkit etc). Several research efforts are already underway that build on or take advantage of the paradigm, including the DAML-S/OWL-S program [28, 29 & 30] and automata-based models for Web Services [31 & 32]. But there is still a long way to go, especially given the apparent long term goal of enabling the automated discovery, composition, enactment and monitoring of collections of Web Services working to achieve a specified objective. A kind of middle ground is also emerging, which provides abstract \signatures of Web Services that are richer than WSDL but retain a declarative aver. Most popular here is the use of automata-based descriptions of permitted sequencing patterns of the Web Services, with a focus on either activities performed [32] or messages passed [33].

Athman et al., [34] proposed Multichannel Adaptive Information Systems [MAIS]. The project MAIS aims at creating a platform, a methodology and a set of design tools to develop distributed information system based one-service. In MAIS, a service is described by a name, a short description, a service category and an aggregation of three types of elements: a Channel (containing contextual information), one or more Service Providers and a Functional Description. When requesting a composite service, a user species the composition requirement and desired QoS constraints, while services are selected based on QoS constraints and their contexts.

Quality of Service (QoS) has been widely used in middleware and networking communities [23 & 36]. The main focus of research efforts in these communities are on the performance of network and devices. There has been a surge in adapting the QoS concept to Web Services in line with their fast growth. Quality of Service or Quality of Web Service (QoWS) could encompass a number of quantitative and qualitative parameter (non-functional properties) that measures the Web Service performance in delivering its functionalities. The researchers [23 & 36] present taxonomy of QoWS parameters to clearly identify the different quality aspects of Web Services.

Efficient access to Web Services as the Web is moving from a Data Web to a Service Web, it is expected that tomorrow's Web will be the repository of a large number of Web Services provided by third-party providers. In that context, the ability to efficiently access Web Services is poised to become of prime importance [37]. In the simplest scenarios, accessing Web Services would consist of invoking their operations by sending and receiving messages. Medjahed et al., in [38] reviewed Web Service technologies from a business-to-business (B2B) application perspective. It outlines the major features of Web Services and examines how they could fit into the B2B interaction environment. Service composition is presented as a powerful tool to combine inter-enterprise applications and enhance the B2B interactions. Papazoglou et al., [39] provide a review of the Web Service technologies as an application of Service-Oriented Computing. They examine several major features of Web Services. They also discuss how the Web Service features benefit the Service-oriented architecture.

As a step further in this direction, the ongoing work in the context of the SELF-SERV project aims at providing high-level modelling constructs and supporting tools to search, compose, execute, monitor, an evolve Web Services. SELF-SERV provides a framework in which Services can be declaratively composed and the resulting composite Services can be executed in a peer to-peer way within a dynamic environment. One of the main objectives of the project is to devise novel integration techniques that allow fast development of new Services from existing ones.

In the next section various Web Service Life Cycles proposed by various renowned researchers are analyzed and compared.

## III. Analysis and Comparison of various Web Service Life Cycles

In this section Web Service lifecycle from beginning of Web to modern Web is discussed. Use, need and purpose of the Web are constantly changing from its beginning to till date. Further, Web lifecycle and Web Service lifecycles activities has been affected and changed accordingly. The researchers proposed the different Web Service Life Cycle Activities according to the requirement of that era. It describes the life of a software product from its beginning to its implementation, delivery, use and maintenance [2]. A traditional software development lifecycle mainly consists of seven phases: planning, requirements analysis, systems design, coding, testing, delivery and maintenance. These phases are originated from the phases of software engineering [8]. It describes the life of a software product from its requirement, to its design and implementation and maintenance [9]. Based on this, a Web Service lifecycle consists of the start of a Web Service, the end of Web Service and its evolutionary stages that transform the Web Service from the start to end.

### Approaches used for Composite Web Services:

There are two basic approaches that are used to create Web Services from scratch. The first approach is termed 'bottom-up'. This is where the code that implements the operations performed by the Service is written first. Then the XML description Web Service Description Language (WSDL) of the service is produced and published in UDDI. The Service code is placed inside a 'container' that provides the required interface for messaging (SOAP or something similar). The second approach is to produce the XML description of the service before it is implemented. So the service is fully described in terms its internal processing, its request and response. This WSDL specification is then used as a guide to writing the code that implements the service [42]. This approach is termed 'top-down'. These two approaches are illustrated in Fig 2.

10.15680/ijircce.2014.0212016

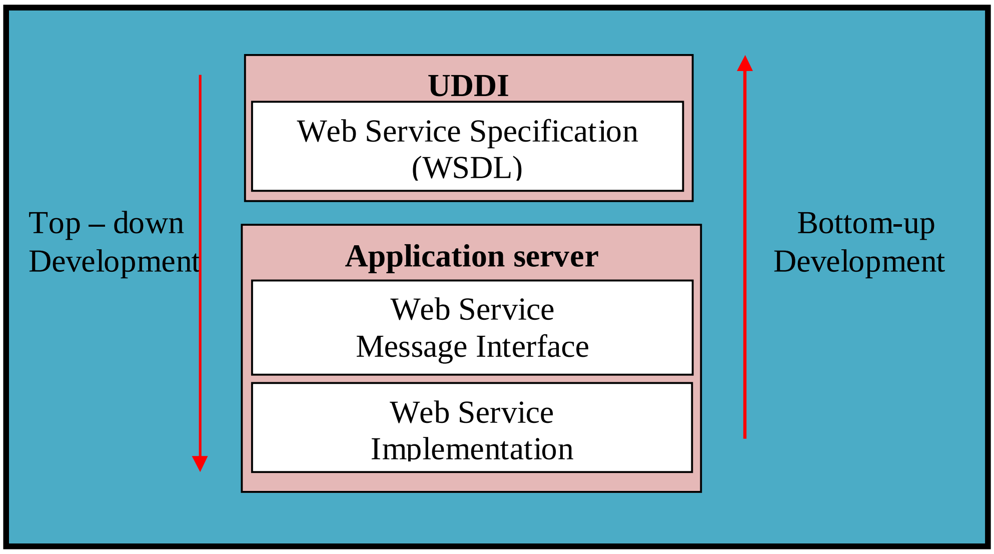
UDDI
Web Service Specification (WSDL)
Application server
Web Service Message Interface
Web Service Implementation
Top – down Development
Bottom-up Development

**Fig 2: Bottom-up and Top-down Web Service Development**

A composite Web Service is an aggregation of multiple Web Services, which interact with each other according to a composition schema. The stages from evolution of Web Service Life Cycle to the modern era are discussed below.

**First Stage:** First stages for Web Service life cycle that will be including the two phase client and server (see Fig. 3). In this Stage was only text information was passed from server to client. And no any another remote machine is required. In this stage the core protocols like XML, HTTP and URI were used. Main issues are for this stage was use of the images, sound, video. Not Asses another machine that will be work for particular machine.

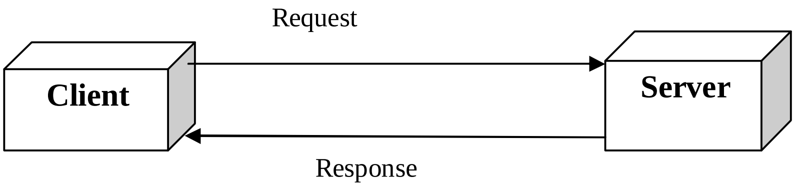

**Fig 3: Composite Web Service life Cycle - First Stage**

**Second Stage:** In second stage the system is enabled to add the graphics. These are organized access to the information on the Internet using Web Service and thus, a user can view the source from multiple clients. Different researchers have contributed in developing Web Service lifecycle in the Web Service community for stage2 [2] (see Fig. 4). For example, Leymann [10] discusses a lifecycle of a Web Service based on explicit factory-based approach, in which client uses a factory to create “an instance” of a particular kind of service; the client can then explicitly manage the destruction of such an instance or it can be left to the Grid environment. Sheth [11] proposes a semantic Web process lifecycle that consists of Web description (annotation), discovery, composition and execution or orchestration. Wu and Chang in [7] consider service discovery, service invocation and service composition as the whole lifecycle of Web Services [7]. Zhang and Jeckl in [12] propose a lifecycle for Web Service solutions that consists of Web Service modeling, development, publishing, discovery, composition, collaboration, monitoring and analytical control from a point of view of Web Service developers [12]. Kwon in [13] proposed a lifecycle of Web Services consisting of four fundamental steps: Web Service identification, creation, use and maintenance.

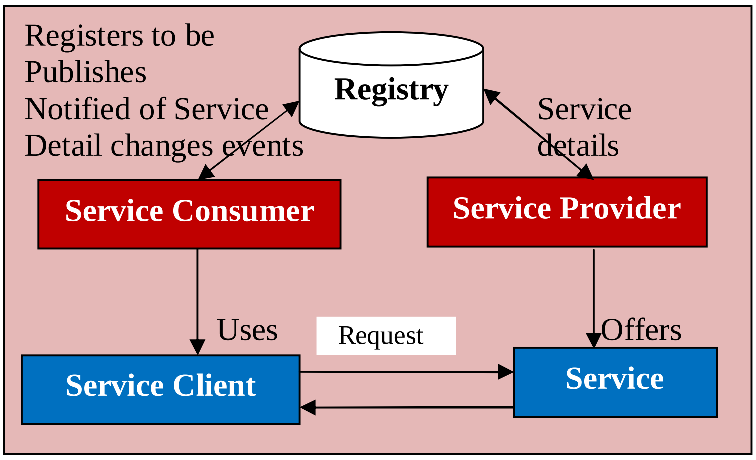

**Fig 4: Composite Web Service Life Cycle- 2nd Stage**

**Third Stage:** The third stage added some significant properties in working with Web Services. In this stage the text became editable to the end user. And the services enabled clients to share the audible data and executable file. The researchers have added a tremendous effort in working with these real time features of stage3. Narendra and Orriens [14] consider a Web Service lifecycle consisting of Web Service composition, execution, in progress adaptation, and re-execution etc. Sun Microsystems considers the lifecycle of Web Services consisting of four stages: design/build, test, deploy/execute and manage [16], which can be considered a model for Web Service developers. Further, demand is an important factor for market and economy development [17]. Fig 5: Composite Web Service life cycle 2008 use for this phase and ASP.NET based this stage.

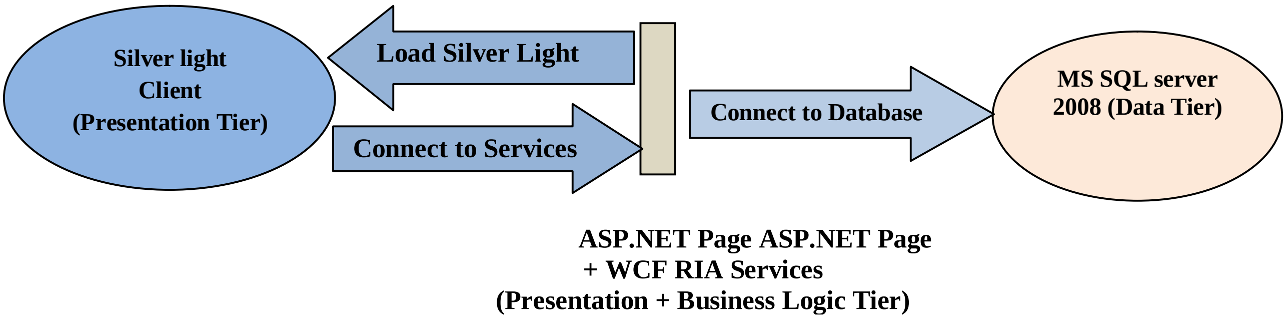

**Fig 5: General Architecture of Composite Web Service life cycle – Stage Three**

Tsalgatidou and Pilioura [15] propose a Web Service lifecycle that consists of two different layers: a basic layer and a value-added layer. The basic layer contains Web Service creation, description, publishing, discovery, invocation and unpublishing, all of these activities are necessary to be supported by every Web Service environment. The value-added layer contains the value-added activities of composition, security, brokering, reliability, billing, monitoring, transaction handling and contracting. Environment are change that user requirement is also change. User requirement is change that is Web Service change and different people are different thinking different model is prepared.

**Fourth Stage:** The Stage 4 added concept of security to the Web Service Life Cycle. Previously the data through Web Services was moving freely without any abstraction from the real world. This life cycle added the security feature that helped making data prone to attacks. The Service section in this life cycle has been further divided internally to make the different processes work corresponding to their Web Service. This also helped in reducing the load on a single Web

10.15680/ijircce.2014.0212016

Service. With all the security efforts and modeling this stage still faced the problems like service Validation and service Construction. They also explore technical challenges related to each activity in Web Service lifecycle. The life cycle of activities related to composite services is illustrated in Fig 6.

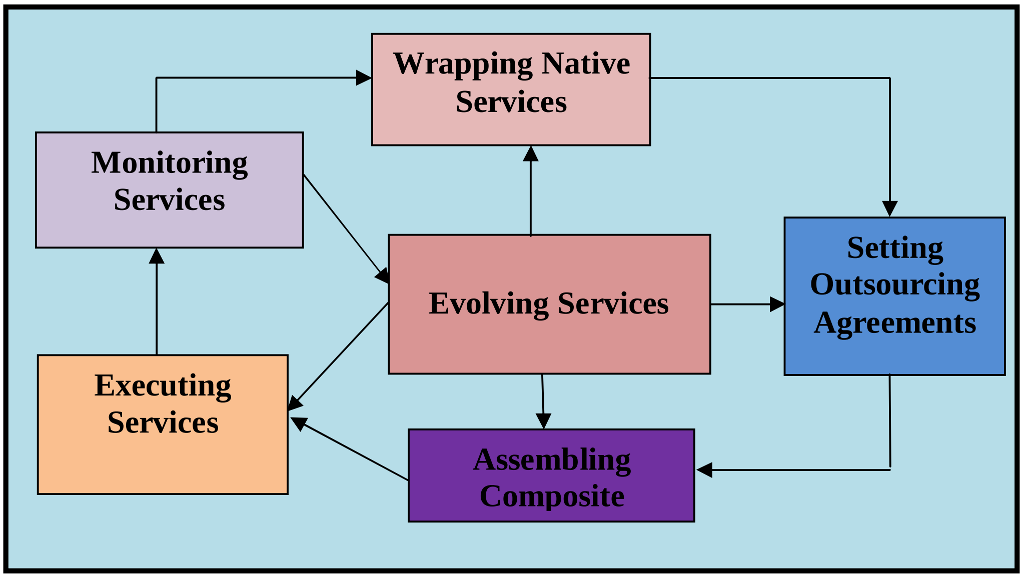

**Fig 6: Composite Web Service Life**

Briefly stated, these activities are:

- **Wrapping Native Services:** ensuring that a native/proprietary service (e.g. legacy application) can be invoked by other Web Services regardless of its underlying data model, message format and interaction protocol.
- **Service Advertisement/discovery:** generating service descriptions and publishing these descriptions in registries for subsequent discovery.
- **Setting Outsourcing Agreements:** negotiating, establishing and enforcing contractual obligations between partner services.
- **Assembling Composite Services:** identifying services to realise a given composition, specifying their interactions at a high level of abstraction and deriving external descriptions and service level agreements for the resulting composite services.
- **Executing Composite Services:** enacting composite service specifications w.r.t execution models satisfying certain practical constraints (e.g. efficiency, availability).
- **Monitoring Composite Service Executions:** supervising composite service executions (e.g., logging service invocations, state changes and message exchanges) in order to detect contract violations, measure performance and predict exceptions.
- **Evolving Services:** adapting composite services to accommodate organizational changes, to take advantage of new technological opportunities or to take into account feedback from monitoring.

With all the affords to make the stage 3 a handy tool to send complex data structures still there was a grave concerned issue of security.

**Fifth Stage:** The overall contribution of this stage was to add validation and service construction for each process that was going in Web Service (see in Fig7). For example if there is need to add a validation to particular object of the class, only that object was working with that validation and there was no impact on others. Previously the validation was restricted to a class.

10.15680/ijircce.2014.0212016

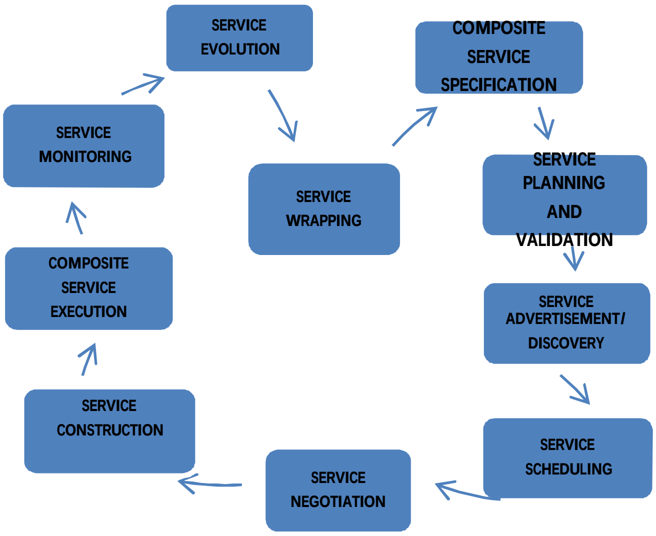
SERVICE EVOLUTION
COMPOSITE SERVICE SPECIFICATION
SERVICE MONITORING
SERVICE WRAPPING
SERVICE PLANNING AND VALIDATION
COMPOSITE SERVICE EXECUTION
SERVICE ADVERTISEMENT/ DISCOVERY
SERVICE CONSTRUCTION
SERVICE NEGOTIATION
SERVICE SCHEDULING

**Fig7: Composite Web Service Life Cycle Stage 5**

Objects in hierarchy can have their own service construction section to handle their respected data. To enable interaction between services among the objects there is a separate negotiation section that helps in co-coordinating the work flow of Web Services.

Different parties have different demands for Web Service life cycle. Therefore, what are the demand- driven Web Service life cycle from the viewpoint of Web Service provider, brokers and requesters respectively. The decrease of demand is an implication for economic recession, as happens in the current global financial crisis. Different parties generally have different demands for Web Services; different demands have also different Web Service lifecycles. Therefore, what are the demand-driven Web Service lifecycle from the viewpoint of Web Service providers, brokers and requesters respectively?

Efforts are going on in making Web Services more dynamic and adaptable. The below given table 1 provides an analytical base for future Web Service Stages on basis of following parameters:

**Table 1**

| | 1st stage | 2nd stage | 3rd stage | 4th stage | 5th stage |
|---|---|---|---|---|---|
| **Content Management** | X | X | Yes | Yes | Yes |
| **Storage** | X | X | X | X | X |
| **Assembly** | X | X | X | Yes | X |
| **Archive** | Yes | X | X | X | X |
| **Compare** | X | X | X | X | Yes |
| **Merge** | X | X | X | X | Yes |
| **Update** | X | X | X | X | Yes |
| **Static Document** | YES | X | X | X | X |
| **Dynamic Document** | X | X | YES | YES | YES |
| **Heterogeneous Information** | X | YES | YES | YES | YES |
| **Security** | X | X | X | YES | YES |
| **Validation** | X | X | X | X | YES |

**Content Management**: - To handle various types of contents with different data types.

10.15680/ijircce.2014.0212016

**Storage**: - Ability to store the raw data and calculated results in the data structures.
**Assembly**: - How different Web Services are assembling the data and information?
**Archive**: - To know about the previously accessed data.
**Compare**: - Comparing the results among the Web Services.
**Merge: -** Merging the results of one or more Web Service.
**Update**: - Ability to update the calculated contents.
**Static Document**: - The content on Web is changeable through Web Services.
**Dynamic Document**: - The Content can be updated dynamically.
**Heterogeneous Information**: - Different type data includes graphics and animations etc.
**Security**: - Provides abstraction to the secured data.
**Validation**: - Validation of individual Objects in a class for their properties.

## IV. Conclusion

The number of clients on Internet is growing with a tremendous speed. To meet the demands Web Services has been added for efficient retrieval of content. Complexity reached to zenith when the graphics and multimedia got incorporated into Web. For handling these data structures in real time Web Services has come into existence to meet the demands. This paper presents the analysis of stages through which Web Services changed to generation next. Also there is broad description about how heterogeneous data is handled by Web Services. The current Web Services technology is creating a bottle neck like situations against growing speed of Web access. To handle huge amount heterogeneous data there is need to further development of Web Services. In the coming paper we will propose a new Life Cycle Model.

## REFERENCES

1. Casati F., M. Sayal & M-C Shan, "Developing E-Services for Composing E-Services", in the proceedings of 13th CaiSE conference, Switzerland.
2. A. Bucchiarone & S. Gnesi, "A Survey on Services Composition Languages and Models", in the proceedings the International Workshop on Web Services – Modeling and Testing (Web Service-MaTe 2006), Palermo, Sicily – June 9, 2006, pp. 51- 63.
3. H. Kuno, M. Lemon, A. Karp & D. Beringer, "Conversations + Interface = Business Logic", Proceedings in the 2nd VLDB-TSE Workshop, Rome.
4. Hofstede, A. T. & Nieuwland, E. "Task Structure Semantics through Process Algebra", Software Engineering Journal 8(1), 1993.
5. Z. Sun, and S.K. Lau, "Customer experience management in e-Services," In Lu, Ruan, and Zhang, Eds, E-Service Intelligence: Methodologies, Technologies and applications, Springer, Berlin, 2007, pp. 365-388.
6. R.T. Rust & P.K. Kannan, "E-Service: A new paradigm for business in the electronic environment", Comm. of the ACM, 46(6) 2003, pp. 37-42.
7. C. Wu & E.A. Chang, "Conceptual Architecture of Distributed Web Services for Service Ecosystems", In Proc. 18th International Conf. on Computer Applications in Industry and Engineering (CAINE 2005), pp. 209-214.
8. R.S. Pressman, "Software Engineering: A Practitioner's Approach (5th edn)", Boston: McGraw-Hill, 2001.
9. S.L. Pfleeger & J.M. Atlee, "Software Engineering: Theory and Practice (3rd Edn)", Pearson Education, Inc. 2006.
10. F. Leymann F. "Web Services: Distributed Applications without Limits: An Outline", in the proceedings of Database Systems For Business, Technology and Web BTW, Leipzig, Germany, Feb. 26 – 28, 2003, Springer, 2003.
11. A. Sheth, "Semantic Web process lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration", Invited Talk, WWW 2003 Workshop on E-Services and the Semantic Web, Budapest, Hungary, May 20, 2003.
12. L.J. Zhang & M. Jeckle, "The next big thing: Web Services composition", In Jeckle and Zhang, Eds, ICWS-Europe, LNCS 2853, Berlin: Springer, 2003, pp. 1-10.
13. O.B. Kwon, "Meta Web Service: building Web-based open decision support system based on Web Services", Expert Systems with Applications vol. 24, 2003, pp. 375–389.
14. N.C. Narendra, and B. Orriens, "Requirements-driven modeling of the Web Service execution and adaptation lifecycle", In Madria S. et al, Eds, ICDCIT, LNCS 4317, 2006, pp. 314- 324.
15. A. Tsalgatidou, T. Pilioura , "An Overview of Standards and Related Technology in Web Services", Distributed and Parallel Databases, 12, 2002, pp. 135-162.
16. Sun Microsystem, "Web Services life cycle: Managing enterprise Web Services", October2003, "http://www.sun.com/software"sun.com/ software, accessed on 24 Dec 2008.
17. J. Jackson & R. Mclver, Microeconomics (7th Edn). Australia. McGraw-Hill, 2004.
18. R. Ladner, et al., "Soft computing techniques for Web Service brokering", Soft Computing, 12, 2008, pp. 1089-1098. voice – 228-688-4679
19. C. Bussler, "The Role of B2B Protocols in Inter-Enterprise Process Execution", in the Proceedings of 2nd VLDB-TES Workshop, Rome, 2001.
20. Alonso G., Casati, F. Kuno, H., Machiraju, V., "Web Services: Concepts, Architecture, and Applications", Springer, Berlin Heidelberg New York (ISBN: 3540440089) (2003).
21. Langdon C.S., "The state of Web Services", IEEE Computer Society, 36(7), 93–94, (2003).
22. Vaughan-Nichols & S.J.:Web Services: beyond the hype. IEEE Computer Society 35(2), 18–21 (2002), ieeexplore.ieee.org/iel5/2/21180/00982908.pdf
23. Langdon, C.S, "The state of Web Services", IEEE Computer Society 36(7), 93–94 (2003).

10.15680/ijircce.2014.0212016

24. W3C:"Simple Object Access Protocol (SOAP)", http://www.w3.org/TR/SOAP/ (2003).
25. W3C: "Web Services Description Language (WSDL)", http://www.w3.org/TR/wsdl (2003).
26. W3C: "Universal Description, Discovery, and Integration (UDDI)", http://www.uddi.org (2003).
27. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic composition of e-Services that export their behavior", In the proceddings of 1st International Conference on Service Oriented Computing (ICSOC), volume 2910 of LNCS, pages 43{58, 2003}.
28. OWL Services Coalition. OWL-S: Semantic markup for Web Services, The OWL Services Coalation November 2003. http://www.daml.org
29. S. A. McIlraith, T. C. Son and H. Zeng, "Semantic Web Services. In IEEE Intelligent Systems", March/April 2001.
30. M. Gruninger, "Applications of PSL to semantic Web Services",In Proceedings of SWDB'03, The first International Workshop on Semantic Web and Databases, 2003 PSL is a project ISO 18629 within the International Organisation of Standardisation.
31. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic composition of e-Services that export their behaviour", In the proceedings 1st International Conference on Service Oriented Computing (ICSOC), volume 2910 of LNCS, pages 43{58, 2003}.
32. R. Hull, M. Benedikt, V. Christophides, and J. Su, "E-Services: A look behind the curtain", In Proc. ACM Symp. on Principles of Database Systems, 2003.
33. Andrea Maurino, Enrico Mussi, Stefano Modaeri and Barbara Pernici, "The MAIS framework for Composite Web Services", International Journal of Interoperability in Business Information Systems, 6:32{64, 2007.
34. Athman Bouguettaya, Quan Z. Sheng, and Florian Daniel, editors, "Advanced Web Services", Springer, 2013.
35. Aurrecoechea, C., Campbell, A., Hauw, L.: "A Survey of QoS Architectures", ACM/Springer Verlag Multimedia Syst. J. 6(3), 138–151 (1998).
36. Marchetti, C., Pernici, B., Plebani, P., "A quality model for multichannel adaptive information", In: WWW04. NewYork,USA (2004).
37. Ouzzani, M., Bouguettaya, A., "Efficient access to Web Services", IEEE Computer Society. 8(2), 34–44 (2004).
38. Medjahed,B., Benatallah,B., Bouguettaya, A.,Ngu,A.H.H.,Elmagarmid & A.K.: "Business-to-business interactions: issues and enabling technologies", VLDB J. 12(1), 59–85 (2003). DOI http://dx.doi.org/10.1007/s00778-003-0087-z
39. Papazoglou M.P. & Dubray, J.:, "A Survey of Web Service technologies", Technical report DIT-04-058, University of Trento (2004).
41. J. Jackson & R. Mclver, Microeconomics (7th Edn), Australia McGraw-Hill, 2004.
42. R.S. Pressman, "Software Engineering: A Practitioner's Approach (5th edn)", Boston: McGraw-Hill, 2001.
43. Neil Simpkins, Implementing a Simple Web Service, T320 E-business technologies:foundations and practice Block 3 Part 1 Activity 5.
44. Xiang Gao,"Planning and Specifying the Composition of Web Service", INFOLAB Tilburg university The Netherlands, htttp://iceb.nccu.edu.tw/proceedings/2003/ DATAS /ID245.pdf.

10.15680/ijircce.2014.0212016