Service Evolution: A Literature Survey

Abstract: In service oriented enterprises, functionalities are provided as self-describing and platform independent services. A service combined with another service, to deliver a larger functionality, is termed as business process. Services undergo changes to adapt themselves to the growing market trends and technologies. This paper presents a literature survey of the existing research being carried out with respect to service evolution in service oriented applications. We classify this research work in five areas: Change patterns in services, Change proneness in services, Change in SOA artifacts, Change in data in SOA and Service Composition Languages extensibilities. We present a comprehensive study comprising of the change problems addressed, the corresponding approaches used to handle them and the future work which could be carried out in each category. This paper is an effort to survey those research areas which are narrowly explored. Hence, this study provides the areas for future work for those who have interest in service evolution.

Keywords: Service, Business process, Service evolution, Change.

Rachna Kohar
Research Scholar,
JNU, New Delhi
E-mail: rachnajnu@gmail.com

I. INTRODUCTION

A service is a loosely-coupled and platform-independent business functionality which achieves a specific business goal [1,2]. Service oriented architecture is an architectural style which guides the creation and usage of services. One of the important aspects of SOA is that the implementation of a service is independent of the interface which exposes the functionalities which are provided by service provider. Fig. 1 shows its architecture. Its major components are service consumer, service provider and UDDI registry. A service consumer invokes the service. A UDDI (Universal Description, Discovery and Integration) registry is a central repository based on XML which enables a service provider to register and a service consumer to locate web services. A service is expressed using WSDL (Web service description language) which is based on XML. It is used to describe the web service including operations supported, messages needed for communication, binding and address to which messages should be sent. SOAP is the standard communication protocol which has XML based format for the exchanged messages between service provider and consumer. These messages are generally sent via HTTP over internet [1-4].

A service could be a composite service which is also termed as a business process. Service composition

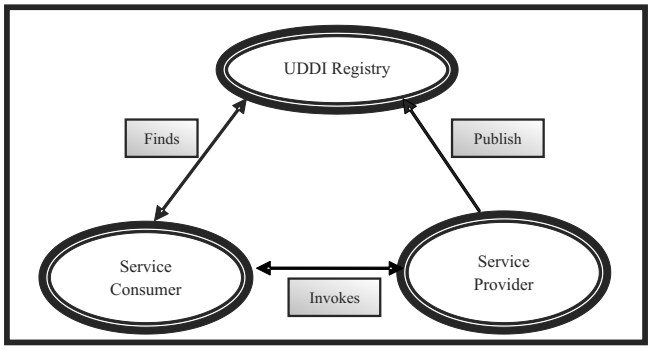

Fig. 1: Service Oriented Architecture

is defined as the process of assembling the existing services to make a composite service/business process. It is usually meant for complex or large applications. It can be achieved in two ways. One is orchestration in which there is a central process which controls and coordinates the services. The other one is choreography in which there is no single process to control the flow of messages between web services. It describes the collaboration of services so as to achieve a common business goal [2, 5-8].

In today's competitive world, there is always a pressure on the service provider to adapt to the changing demands of the market. Therefore, service provider tends to change his service. Evolution in terms of service is the development of service and then updating it [9,10]. We classify the changes at three levels in SOA. These

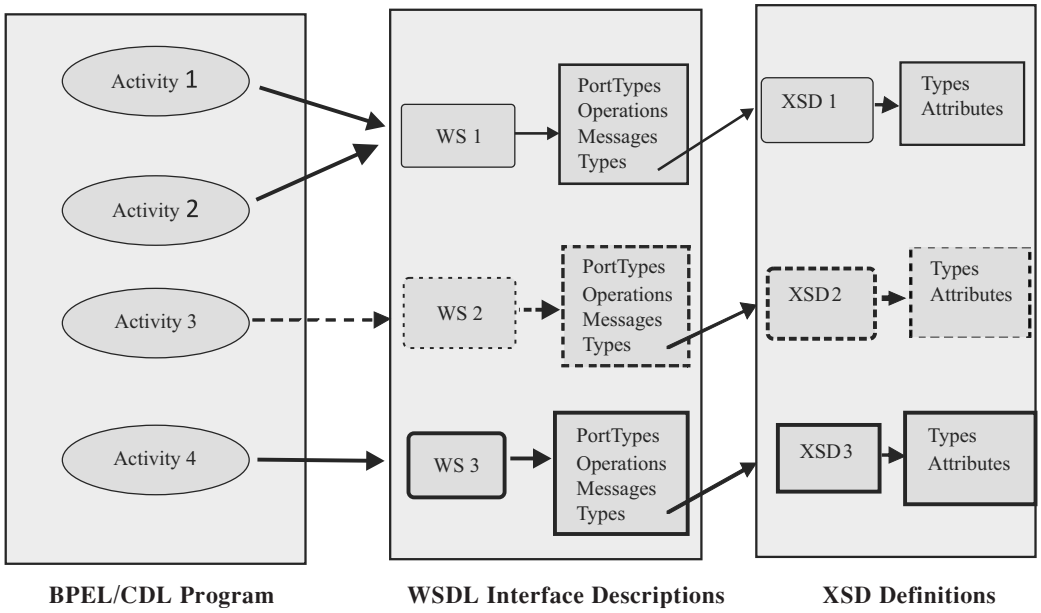

Fig. 2: Example Scenario for Changes in Services

levels are: composition level, service level and data definitions level. Fig. 2 depicts a scenario showing these changes. As we know that a service is composed of many services to achieve specific business functionality, thus, the composite service comprises of many activities to interact with them. These activities are send/receive/assign etc. in BPEL notation and roleType/participantType/informationType etc. in CDL program. These activities may undergo changes such as addition/deletion/modification. The BPEL/CDL program refers WSDL document. A service interface is expressed as a WSDL document which contains operations/messages etc. These service interface elements may undergoes changes like addition/deletion or modification. The service interface uses data type definitions which are expressed in XSD document or may be embedded in the WSDL document itself. The XSD document contains the data type's definitions for the messages which are present in WSDL document. These data type definitions may also undergo changes. Figure 2 shows the deleted activity/service/data type in dashed ellipse/rounded rectangle and the added activity/service/data type in thick-bordered ellipse/rounded rectangle [11-17].

The survey by Yi Wang and Ying Wang [18] addresses the change management lifecycle in a service based environment under four categories: service adaptation, process flexibility, service evolution and change analysis and management. The first category signifies achieving interoperability among interacting services and business protocols by mediating their differences. It is based on the mismatch patterns among services like signature mismatch, message order mismatch. Under second category, extensions of BPEL are studied. For e.g., VXBPEL deals with the variation points and variants in BPEL, the extension AO4BPEL provides notations in BPEL to specify data validation and security as aspects. In the third category, web service versioning management is studied which includes monitoring of changes in service version and notifying them to clients. The last category addresses the change management lifecycle of a business process which includes the study of impact of changes in a private (BPEL) process of one partner on another partner process, problem of managing running instances of a web service when business protocol changes etc.

II. RESEARCH METHOD

As the change management in SOA has vastly emerged, our primary research question is: "What are the areas in which there is scope for the researchers to conduct research?" The primary question is divided into four research questions (RQ) shown in Fig. 3.

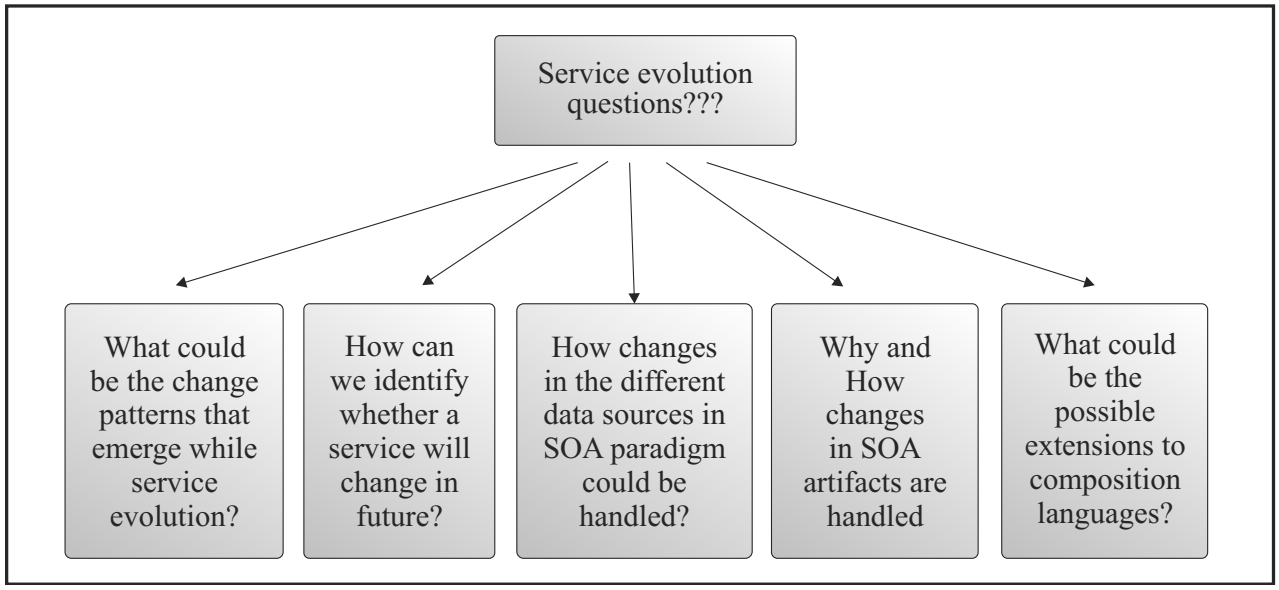

Fig. 3: Research Questions

We present our study for these RQs under below areas.

Change patterns in services: In this area, we focus on the study of the research done by different authors for change patterns in services. Our study covers different kinds of patterns for a service such as Split-Map and Merge-Map and for a business processes such as Activity Merge/Change and Activity Extend/Delete.

Change proneness in services: Under this area, we present the work done in identifying change proneness in services. Authors predict change proneness of services via change indicators like coupling, cohesion or data type complexity. As per our analysis, we find this area as a wide scope of research for the researchers to explore.

Change in data and SOA: Services in SOA uses many service which in turn uses heterogeneous data. This area covers the study of changes and their handling in the data used by services.

Change in SOA artifacts: Various mechanisms have been developed to handle the changes in SOA artifacts like additions or deletions in artifacts such as BPEL program code or WSDL interface descriptions.

Service Composition Languages extensibilities: This area comprises of the literature work for WS-BPEL (a business process language) and WS-CDL (a choreography language) extensions which enhance the capabilities of BPEL.

We have considered four electronics databases as the primary sources for our literature study. These are listed in below Table 1.

Table 1: List of sources

#	Database	URL
1	IEEE	http://ieeexplore.ieee.org/
2	Springer	http://link.springer.com/
3	ACM SIGMOD	http://www.sigmod.org/
4	Research Gate	http://www.researchgate.net//

This paper is organized in three sections. In section 2, we present our research method. In section 3, we present our classification of the research in change management in SOA in detail. In Section 4, we conclude the paper.

III. CATEGORIES IN CONTEXT OF SERVICE CHANGES

This section discusses the work done in context of changes in services. Firstly we review the work done to address different types of dependencies among services. We discuss the change patterns and change proneness in service. Then we focus on the work that is related with managing changes in database in service oriented environment. Finally, we review extensions of BPEL that are not covered in the referred survey paper.

A. Change Patterns in Services

In this section, we discuss the existing research work on change patterns in service based applications. Firstly, we discuss the work done on finding the change patterns in a single service [19,48]. Next, we discuss the research on change patterns in a composite service (business process) [21]. Lastly, we discuss the work done by various authors to determine the patterns of changes in services at the architecture level [22,23].

A service go through various types of changes like addition, deletion etc. forming certain patterns. Service evolution patterns [19] are the reusable strategies for the services to resolve the issues while it goes through evolution and to estimate the impact of changes on service consumers. This is achieved after understanding the patterns which arise while changes are done in service. This is done with the help of a service evolution model which uses the concept of service dependency on consumers. Based on the analysis of these dependencies among service and consumer, the service evolution patterns are proposed. The proposed patterns are: compatibility, transition, split-map and merge-map. The first pattern showcases that for changes, some service consumers are compatible and some are not. The second pattern helps the service consumer for smooth transition to the changed service. Third pattern is concerned with the frequency of changes that are applied on operations of services. This assists to split more frequent operations form the less-frequent operations. The last pattern is helpful in determining the services which have common operations to enable merging between the services.

Now, we discuss the work done for the change patterns in business processes. As we know that a business process or a composite service comprises of many activities such as control flow, sequence etc. which can be changed. For example, an activity 'A' split into multiple activities 'A1', 'A2' … and 'Am' that are to be executed in parallel. The process change patterns are activity split/change, activity merge/change and activity extend/delete [20]. A process designer for BPEL is developed to provide templates which supports process change patterns. These patterns are applied by user via the process designer wizard without re-designing the process model manually. This helps to identify which version of the process should be selected and executed at run time using abstract process execution.

Change patterns in architecture-centric evolution for services in SOA are proposed for re-use [21,22].The sequential architectural changes are analyzed from the architecture change logs to identify patterns. Based on these, pattern-based evolution model is proposed by the author is 4-tuple PatEvol =<SArch,OPR,CNS,PAT>, where SArch is the architecture element to which a change pattern can be applied , OPR refers to change operator, CNS refer to a set of pattern specific constraints and PAT represents a recurring composition of change operationalization on architecture elements. This model helps the architects for reuse in architecture-centric evolution of services.

B. Change Proneness in Services

The research work which has been done in identifying the change proneness of a service, by different authors, is discussed here. As the service evolves from time to time, there is a need to determine the quality and stability of the service. The service provider does not know in advance the impact of changes on the service consumers which makes the authors feel the necessity to determine change proneness of a service [47]. This research helps the developers to design the service towards more stability and quality.

In [23], indicators of change such as heuristics, anti-patterns, cohesion, metrics etc. are proposed to predict the change prone Java APIs. Based on these indicators, Java APIs are classified into more change-prone and less change-prone. The research is intended to extend to investigate the change proneness of service interfaces, by taking the work done for java APIs as the base because service interfaces are mapped to these APIs. The change-proneness of a WSDL interface is identified using metrics i.e. Cohesion metrics: Service Interface Data Cohesion metric, Data Type Cohesion metric and Data Type Complexity: Complexity Based Types metric, in [24].The metrics are empirically calculated for the fine grained changes between subsequent versions of real time services. Using this info, service developers are assisted to develop less change-prone interfaces for a service. It has been concluded that increased complexity and less cohesion leads to more change-prone interfaces.

In [25], the impact of service patterns and anti-patterns on the maintenance and evolution of service based systems is studied. As specified by the author, the frequently changed services face some recurring design problems during the maintenance process. While proposing solutions, some bad practice solutions known as anti-patterns in service based systems come into picture. For e.g., bloated service which has operations that are less cohesive in nature and are less reusable and incur high maintenance cost. Metrics such as number of changes and size of changes (i.e., code churns) performed by developers, are used to measure the change-proneness of a service. The observation by the author states that the metrics values involved in a service pattern is less than the metrics value in the service anti-pattern. Thus, services containing a high number of antipatterns (respectively patterns) are likely to change more.

From the above discussion, we come to know that the current research in this area is very limited. There is much scope for more research. The work done so far in this area does not comprise of coupling metrics, granularity and reusability metrics etc. that may result in predicting the change proneness of services. In addition, the type of changes i.e. functional or non-functional can be considered to have a more comprehensive prediction.

C. Change in Data/Database in SOA

SOA combines business data and process to develop services. The two areas data and SOA are related to each other [26,27]. The data-oriented SOA aims to remove the complexities in accessing heterogeneous data for different services. This arises the need to emerge data services for centralized and consistent view of data and also to resolve data issues like management, quality and movement of data among services [28,29].

In this section, we shall discuss about the research work which has been done for the detection and handling of changes in data and databases in a SOA paradigm.

As the database systems evolved, the need to incorporate the service concepts also arose. In [30,31], an architecture called Service Oriented Database Architecture (SODA) is described which was developed for SQL Server. It contains SOA features built in the database engine. SODA consists of many features and the one which is related to the data changes is a feature of change notifications in the database in a SOA paradigm. The service and data logic can be on a single machine or can be on different machines. Whenever a change in data occurs, that change should be propagated to the client who is accessing the data. SODA has an integrated database change notifications (DCN) feature that notifies the client when a change occurs in the underlying database.

A composite service comprises of many component services and each of these may require data from different data sources. To handle the heterogeneous data, Service Data Object (SDO) is used [32].A SDO contains three concepts: Data Object (DO), Data Graph and Data Access Service (DAS). DO is nothing but data (business). Data Access Service (DAS) layer provides access to data source and creates data graph. The changes in data are depicted in data graph. DAS applies changes in data graphs whenever data in data sources changes.

We observed that in the above discussion, very little research is done so far which is related to data in service oriented systems. More techniques can be used to explore the change management of data in service based systems.

D. Changes in SOA Artifacts

In this section, we discuss that how changes in different service artifacts are handled in SOA.

In [33], service design process is shown which comprises of many stages such as business componentization (business objectives are set), service discovery (suitable service as per business objectives are discovered) etc. At each stage, artifacts are maintained. Few examples of artifacts are business goal, service specification, metrics, KPI etc. Changes at any stage of service design process may affects related artifacts of other stages. This may result in any of the three changes in artifacts: addition/deletion/modification of artifact. These changes are analyzed and propagation analysis is done for the artifacts which impacts other related artifacts.

The service artifacts are classified in [34] as design artifacts like use-case and sequence diagrams and code level artifacts like BPEL Code, configuration files etc.

The changes in artifacts at one level affects artifacts at another level in a SOA based solution. The author has developed a framework Morpheus to model semantics based change relationships between artifacts at different levels.

BPEL program code, WSDL service interface descriptions and XSD data type definitions are the main service artifacts [35]. The SOA applications are heterogeneous in nature and thus, the maintenance may require expert knowledge to coordinate changes. The BPEL program references WSDL service interface descriptions and XSD data type definitions which in turn uses XSD data type definitions. The correct understanding of the dependencies between these artifacts is required for a maintainer to incorporate change effectively. This is done using rule based system which uses expert reasoning on these artifacts.

E. Service Composition Languages Extensibilities

BPEL is a standard language to orchestrate partner services to form a service composition i.e. business process. To increase the flexibility of business processes to adapt to various concepts, extensions of BPEL are introduced. We now discuss the extensions of BPEL. In [36] BPELCHor is introduced, an extension to BPEL, which describes the choreography using BPEL notations. The extension basically includes participant behavior descriptions which define the control flow dependencies between activities and participant topology which defines structural aspects of choreography by specifying participant types, participant references, and message links. BPEL has limited reusability for the processes due to the strong coupling between WSDL interfaces and BPEL. Thus, the BPELLight was proposed in [37] to develop WSDL-Less BPEL i.e. describes the interaction between two partners without dependency on WSDL. This enables BPEL to define interfaces of service using any Interface Definition Language (IDL). BPEL4People[38] introduces information about users who participates in process, in addition to information such as variables and partner links in BPEL process. It helps to identify the different ways that people interact with processes and how the process identifies the people to interact with. The people activity, like other BPEL activities, has a name, input and output to specify the data exchanged with the task. BPEL for interoperable pervasive computing is developed by the author in[39]. It allows users in wired computing settings to model the applications used in the networks but is not capable to model dynamic wireless networks. A lightweight BPEL Engine, Silver, is developed for the mobile devices.

WS-CDL is used to describe the choreography of services i.e. sequence of interactions between services [43-46,49-50]. There are various extensions which have been proposed for WS-CDL so that it can be made flexible to incorporate more features. We now discuss the extensions of WS-CDL. CDLExt is a model extension of WS-CDL which is proposed in [40]. The main purpose of the extension is to manage the service layer including the dependencies among services and IT artifacts such as business, database etc., managing the non-functional attributes of a service etc.. This helps the architects to decide whether the service with given parameters would deliver under a specific contract. [41] extends WS-CDL to make the sub choreography elements such Exception, variable etc. within the main choreography element reusable. This is achieved by making choreography templates to make them reusable. The reusability feature makes the interaction patterns of services available a generic entity in a composition. [42] provides an approach to build QoS aware service based processes. The choreography of services are annotated with SLA references because typically non-functional requirements are described using SLAs among the interacting services.

IV. CONCLUSIONS

This paper provides a more comprehensive review of the research work that is being carried related to changes in SOA. Firstly, we have presented a summary of the literature which is covered in the survey by Yi Wang et al. Then, we have classified the research work done in change management under five areas: Change patterns in services, Change proneness in services, Change in SOA artifacts, Change in data in SOA and Service Composition Languages extensibilities. We have discussed the work done in the research and also figured out the issues that are still open and remain unexplored in each of the category.

Fig. 4 shows the percentage of research, as per our study, that has been carried out in each of the five areas of the existing research. We can see that ,relatively, last area which is “Service Composition Languages

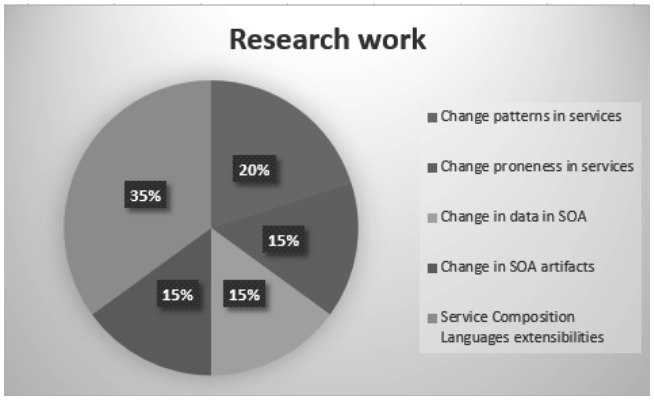

Fig. 4: Amount of research work done in areas

extensibilities " as depicted in the pie chart is the most explored area by researchers as compared to other areas of the research in this paper.

The amount of research that has been conducted in the five areas over the last 15 years is displayed in Fig. 5. The chart display that the second area "Change proneness in services " has been explored in the recent years which is sign that there is a wide area of scope for research in this area. Another area which is very less explored in the recent years is the third area i.e. "Change in SOA artifacts".

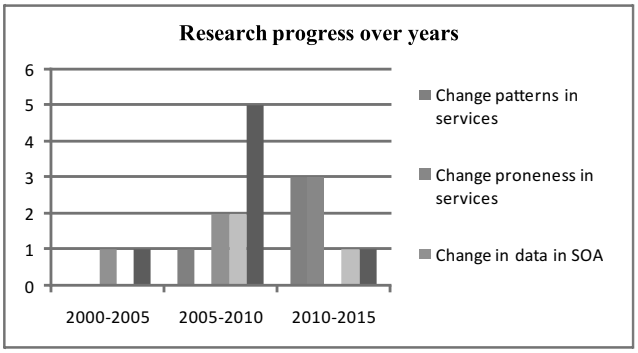

Fig. 5: Analysis of research carried in areas over years

REFERENCES

[1] Erl, Thomas. Service-oriented architecture: concepts, technology, and design. Pearson Education India, 2005.

[2] Papazoglou, Mike P. "Service-oriented computing: Concepts, characteristics and directions." Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on. IEEE, 2003.

[3] Perrey, Randall, and Mark Lycett. "Service-oriented architecture." Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on. IEEE, 2003.

[4] Erl, Thomas. Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR, 2004.

[5] Krafzig, Dirk, Karl Banke, and Dirk Slama. Enterprise SOA: service-oriented architecture best practices. Prentice Hall Professional, 2005.

[6] Laskey, Kathryn B., and Kenneth Laskey. "Service oriented architecture." Wiley Interdisciplinary Reviews: Computational Statistics 1.1 (2009): 101-105.

[7] Natis, Yefim V. "Service-oriented architecture scenario." (2003).

[8] Channabasavaiah, Kishore, Kerrie Holley, and Edward Tuggle. "Migrating to a service-oriented architecture." IBM DeveloperWorks 16 (2003).

[9] Papazoglou, Mike P. "The challenges of service evolution." Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2008.

[10] Andrikopoulos, Vasilios, Salima Benbernou, and Michael P. Papazoglou. "On the evolution of services." Software Engineering, IEEE Transactions on 38.3 (2012): 609-628.

[11] Akram, Salman, et al. "A change management framework for service oriented enterprises." International Journal of Next-Generation Computing 1.1 (2010).

[12] Wang, Yi, et al. "Change impact analysis in service-based business processes." Service Oriented Computing and Applications 6.2 (2012): 131-149.

[13] Romano, Daniele, and Martin Pinzger. "Analyzing the evolution of web services using fine-grained changes." Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012.

[14] Ginige, Jeewani, Uma Sirinivasan, and Athula Ginige. "A mechanism for efficient management of changes in BPEL based business processes: an algebraic methodology." e-Business Engineering, 2006. ICEBE'06. IEEE International Conference on. IEEE, 2006.

[15] Akram, Mohammad Salman, Brahim Medjahed, and Athman Bouguettaya. "Supporting dynamic changes in web service environments." Service-Oriented Computing-ICSOC 2003. Springer Berlin Heidelberg, 2003. 319-334.

[16] Papazoglou, Michael P., et al. "Service-oriented computing: State of the art and research challenges." Computer 11 (2007): 38-45.

[17] Papazoglou, Michael P., et al. "Service-oriented computing: a research roadmap." International Journal of Cooperative Information Systems 17.02 (2008): 223-255.

[18] Yi Wang, Ying Wang, A survey of change management in service oriented environments,: In Service Oriented Computing and Applications-Springer (2013), 259-273.

[19] Wang, Shuying, et al. "Service Evolution Patterns." Web Services (ICWS), 2014 IEEE International Conference on. IEEE, 2014.

[20] Kim, Dongsoo, Minsoo Kim, and Hoontae Kim. "Dynamic business process management based on process change patterns." Convergence Information Technology, 2007. International Conference on. IEEE, 2007.

[21] Ahmad, Aakash, Pooyan Jamshidi, and Claus Pahl. "Pattern-driven reuse in architecture-centric evolution for service software." (2012).

[22] Ahmad, Aakash, Pooyan Jamshidi, and Claus Pahl. "Graph-based pattern identification from architecture change logs." Advanced Information Systems Engineering Workshops. Springer Berlin Heidelberg, 2012.

[23] Palma, Francis, et al. "Investigating the Change-Proneness of Service Patterns and Antipatterns." Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on. IEEE, 2014.

[24] Kalouda, M. Analyzing the Evolution of WSDL Interfaces using Metrics. Diss. TU Delft, Delft University of Technology, 2013.

[25] Romano, Daniele. "Analyzing the change-proneness of service-oriented systems from an industrial perspective." Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013.

[26] Linthicum, David S., Defining, Designing, and Implementing SOA-Based Data Services.

[27] Data Integration in a Service-Oriented Architecture,White Paper, Informatica, 2005

[28] Lawson, J. "Data services in SOA: maximizing the Benefits in enterprise architecture." Oracle,[Online] April (2009).

[29] Mínguez, Jorge, et al. "A SOA-based approach for the integration of a data propagation system." Information Reuse & Integration, 2009. IRI'09. IEEE International Conference on. IEEE, 2009.

[30] Campbell, David. "Service Oriented Database Architecture: App Server-Lite?."Proceedings of the 2005 ACM SIGMOD international conference on Management of Data. ACM, 2005.

[31] Tok, Wee Hyong, and Stephane Bressan. "DBNet: A service-oriented database architecture." Database and Expert Systems Applications, 2006. DEXA'06. 17th International Workshop on. IEEE, 2006.

[32] Resende, Luciano. "Handling heterogeneous data sources in a SOA environment with service data objects (SDO)." Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007.

[33] Zhang, Liang-Jie, et al. "Variation-oriented analysis for SOA solution design."Services Computing, 2007. SCC 2007. IEEE International Conference on. IEEE, 2007.

[34] Ravichandar, Ramya, et al. "Morpheus: Semantics-based incremental change propagation in soa-based solutions." Services Computing, 2008. SCC'08. IEEE International Conference on. Vol. 1. IEEE, 2008.

[35] Goehring, George, et al. "A Knowledge-Based System Approach for Extracting Abstractions from Service Oriented Architecture Artifacts." International Journal of Advanced Research in Artificial Intelligence 2.3 (2013): 44-52.

[36] Decker, Gero, et al. "BPEL4Chor: Extending BPEL for modeling choreographies." Web Services, 2007. ICWS 2007. IEEE International Conference on. IEEE, 2007.

[37] Nitzsche, Jörg, et al. "BPELlight." Business process management. Springer Berlin Heidelberg, 2007. 214-229.

[38] Kloppmann, Matthias, et al. "Ws-bpel extension for people–bpel4people." Joint white paper, IBM and SAP 183 (2005): 184.

[39] Hackmann, Gregory, Christopher Gill, and Gruia-Catalin Roman. "Extending BPEL for interoperable pervasive computing." Pervasive Services, IEEE International Conference on. IEEE, 2007.

[40] Dusza, Konrad, and Henryk Krawczyk. "Managing distributed architecture with extended WS-CDL." Parallel Processing and Applied Mathematics. Springer Berlin Heidelberg, 2008. 281-290.

[41] Mardukhi, Farhad, Naser NematBaksh, and Kamran Zamanifar. "EXTENDING WS-CDL TO SUPPORT REUSABILITY." International Journal on Web Service Computing 2.1 (2011).

[42] Rosenberg, Florian, et al. "Integrating quality of service aspects in top-down business process development using WS-CDL and WS-BPEL." Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International. IEEE, 2007.

[43] http://www.w3.org/TR/ws-cdl-10/

[44] Fredlund, Lars. "Implementing ws-cdl." Proceedings of the second Spanish workshop on Web Technologies (JSWEB 2006). 2006.

[45] Weerawarana, Sanjiva, et al. Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. Prentice Hall PTR, 2005.

[46] Mendling, Jan, and Michael Hafner. "From WS-CDL choreography to BPEL process orchestration." Journal of Enterprise Information Management 21.5 (2008): 525-542.

[47] Bieman, James M., et al. "Design patterns and change proneness: An examination of five evolving systems." Software metrics symposium, 2003. Proceedings. Ninth international. IEEE, 2003.

[48] Weber, Barbara, Stefanie Rinderle, and Manfred Reichert. "Change patterns and change support features in process-aware information systems." Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2007.

[49] Acharya, Amit, et al. Patterns: Implementing an SOA using an enterprise service bus. IBM, International Technical Support Organization, 2004.

[50] Newcomer, Eric, and Greg Lomow. Understanding SOA with Web services. Addison-Wesley, 2005.

❐