I J C T A, 9(28) 2016, pp. 1-12

A SURVEY ON MANAGING CHANGES IN SERVICE BASED SYSTEMS

V. Kavitha* and P. Shanmugapriya**

Abstract: In service-oriented computing paradigm, inter-organizational applications and information systems can be built upon services from different service providers. Services are subject to changes required by the clients, organizational and changes in regulatory policies. The demanding problem of service change management has been studied actively in recent years. This paper provides an extensive overview of the current research on change management, Analysis of change and its impact in the context of service-oriented computing. Service changes are classified into functional and non-functional changes. Then, we review the existing work on change management based on *Service Adaptation, Change Analysis and Management in Service Compositions*. In each category, the discussion is based on focusing changes, the proposed approaches to dealing with the change problems, and the change issues that remain to be solved.

Keywords: Service-oriented computing, Change management. Change analysis, Change Impact, web services.

1. INTRODUCTION

Change management is a conventional problem and an indispensable part in the development and preservation of software based systems, due to the progress nature of computing programs [40]. Change management has been studied in a wide range of research areas: software engineering [35,39,40,50,51,61], distributed systems [21,38,76], database management systems [6,7,27,73,84,86,87], information systems [19,71], and workflow systems [15,17,31,32,62–65,68,69,77]. Closely related to the change management in service-based systems, in the software engineering area, alteration approaches to mediating mismatches between software components are proposed. In addition, for distributed systems and information systems[78,80], ideology and methodology for managing evolutionary changes introduced to system components and module that are not expected at the design time exist. These efforts are valuable and can be leveraged to develop mechanism for the Web service adaptation and the change management for Web service-based systems. Moreover, various type of changes of Web services that belong to different service providers usually happen separately, which may affect Inter-organizational collaboration building on these Web services. The workflow change management, which has been widely studied since 1990s, is also related to the research on the change management for web service-based business process. The focus is put on handling workflow schema changes (static change) and the exodus of work-flow instances (dynamic change) when their schemas change. However, due to the vibrant nature of Web services and the multifaceted relationships between services and business process, these studies are insufficient to fully address the change issue and support the flexibilities for service-based business process.

SOC facilitates the low-cost and swift composition of loosely coupled applications. Inter-organization application can be built-up by integrating dissimilar types of web services from diverse service providers. Due to a range of origins such as governmental policies, ecological conditions, client requirements and technology, services may need to vary from time to time. A service can revolutionize in many aspects

* Research Scholar, Dept. of CSE, SCSVMV University, Asst.Prof. Dept. of CSE, Sri Sai Ram Engineering College, Tambaram, Chennai – 44. **Email:** kavidhaya.phd@gmail.com

** Assoc. Prof, Dept. of I.T., SCSVMV University, Kanchipuram **Email:** priya_prakasam@gmail.com

including its interface, industry protocol as well as QoS attribute. A change of a service interface and an industry protocol may cause incompatibilities between parities. Moreover, a multifaceted service may be supported by no of fundamental business processes [56]; a change to the service frequently requires changes applied to its supporting business processes and to other services the same business processes hold. Service-based application and information system will need to function correctly despite of predictable and unpredicted changes related to web services and business processes. Therefore, when a transform happens, it is important to understand the type of this change, the change crash including the pretentious scope and how deeply it affects the entire system and method to handle it. Due to the distributed and vibrant characteristics of services, change management is a demanding issue in service-based environments [54,56,58].

A allusion architecture also has been developed for managing vibrant inter-organizational business processes which consider a set of functional and non-functional requirements for eSRA that comprises gears on three refinement levels and their information exchanges. The interacting parties must be able to defend their business secrets and the overall performance of an eSRA-acquiescent system must not obstruct business interaction[90].

The end goal of the service change management is to understand change automation in the entire life cycle of web service-based applications. The reason of this paper is to provide an widespread survey of the existing research on change management in service-based environment. This paper, reviews the existing research from the following categories:

Service Adaptation refers to the facility of web services to fine-tune themselves in order to conquer incompatibilities caused by changes in interacting services at the level of interface and/or external behavior. The main idea for service adaptation is about designing adapters/operators to intercede a range of types of differences between Web services. To generate adapters, a number of approaches have been used based on the foremost types of mismatches among service interfaces and/or business protocols.

Service Evolution is the procedure of changing services animatedly and constantly. In this branch of research, strategy and mechanism supporting Web service versioning control and growing service stipulation such as business protocols are recommended.

Changes Analysis and Management in Service Compositions Web services operate in highly vibrant environments. Participant services in a service composition, particularly for long running service compositions, may vary at any time. In the literature, we have seen a number of efforts that provide mechanisms for changes analysis and management in service composition. The center of attention is on detecting, analyzing, propagating, as well as reacting to service changes.

This paper is organized as follows. In Sect. 2, we classify the types of changes that can happen to services into two categories as functional and non-functional changes. In Sect. 3, we review and summarize the methodologies, technique, and approaches projected in the literature for dealing with the various types of change issues. Finally, we conclude this paper in Sect. 4.

2. CHANGES IN SERVICE-BASED ENVIRONMENTS

Services are published, exposed, elected, and incorporated based on their external specifications, which contains the information of signatures, message exchanging constraint and orders, QoS properties, and behaviors of services. Due to a variety of reasons such as industry regulations, environment and client requirements, services need to vary from time to time. A service can vary in many aspects of its external specification, such as schemas of messages, operation granularity, message exchanging sequences, and operation existence etc.

The XML-based Web services specifications: WSDL and BPEL offer standards for relating the interface and behavior of a Web service. Apart from these, we have seen an amount of modeling techniques: process algebra [12,20,49], automata [10,16,18,25], and Petri nets [46–48,75,82-83] in the literature in order to provide prescribed semantics for the behavior of Web services. This paper categorize the changes that can happen to services into the following two broad categories:

1. *Functional changes*. It discuss about the behavior of a Web service ie. how it interact with its users by exchanging messages. Behavioral changes comprise variations relating to service behaviors, for example, changing order of messages in business protocols or removing activities in abstract BPEL processes.
2. *Non-functional changes.* contain disparity of QoS properties. In general, the QoS of a service contains a number of non-functional attributes such as privacy, standing, usability, and performance price, which are important for the potential patrons of this service [44]. Services having analogous functions may have dissimilar QoS parameters. QoS properties are vital criterion in forming service level agreements (SLAs) between business partners. QoS properties of a Web service may vary at any moment. For example, a service provider can adjust the price of its web services at any time.

A service change may be limited locally without any further crash on the entire system; it may also deeply affect the complete value chain and has wave effects as well [56]. Thus, change detection and impact analysis are critical and compound tasks in the change management for service-based applications and information systems.

3. SERVICE CHANGE MANAGEMENT

This section discuss the work done for dealing with service changes in the framework of SOC. According to the above two research categories, we shall firstly review the major approach and methods for adapting service interfaces and protocols, which are significant means for handling convinced types of change problems. Then, we will discuss the approaches planned in the literature for enabling process flexibility and service fruition. Finally, we will focus on the effort for change analysis and management in service-based systems.

3.1 Service Adaptation

Changes can root incompatibilities between interacting services. The existing Web service standards such as SOAP and WSDL can solve the compatibility concern among Web services at the lower level of abstraction, that is, messaging, incompatibilities still subsist at higher levels of abstraction, that is, service interfaces and behaviors. Service adaptation refers to the ability of changing a service itself in order to work together with other services [56]. Service adaptation is a vital means to reach interoperability between services when alterations happen.

Recent research on service adaptation primarily concentrates on adapting service interfaces and business protocols[2]. Adaptation at the interface level is about regulating interfaces in order to conquer differences and incompatibilities among interacting web services. A service protocol describes the preferred message exchange patterns between the service and its clients [9]. Adaptation at the protocol level is about mediating the different types of differences between service protocols. There are two foremost types of approaches to adapting services: adaptation based on mismatch patterns and adaptation based on formal semantics.

3.1.1 Service adaptation based on mismatch patterns

Benatallah et al. [8] confer about scheming patterns for adapters based on a set of disparity patterns for service interfaces and protocols. This research presents a general principle for dealing with web service adaptation based on taxonomy on service differences. Motahari-Nezhad et al. [52,53] presents semi-automated mechanisms for perceiving and designing adapters to mediate some sorts of incompatibilities between services.

Kongdenfha et al. [36] argued about the dissimilarity between an internal service accomplishment and standard external specifications at the interface and protocol level. The business logic is treated as the major concern, and the adaptation logic is specified as crosscutting concerns. The authors intended a set of template to grip the mismatch types: **S**, **O**, **E/M**, and **S/M**. A prototype tool is build up to support the template instantiation and implementation. The mechanism for designing adapters for coping with some of the different types with special consideration, for example, minimal adapters [70], have been recommended. Different from the above work, Ponnekanti and Fox [60] examines the interoperability of independently growing Web services in Web-based applications. The goal is to enable replacement of functionally similar services that are derivative from a common base. They recognize four types of incompatibilities between Web services as structural incompatibility, value incompatibility, encoding incompatibility, and semantic incompatibility. The researcher spotlight on the structural and value incompatibility referred to as SV-incompatibility. The interoperability between Web services then can be comprehend by semi-automatically generated cross-stubs.

In general, adapters are implemented as middleware mechanism that arbitrate two Web services in a service environment. From the above description, we can study that automatic adapter generation for mediating the different types of incompatibilities is a demanding task, particularly for semantic incompatibilities. Relying on types of discrepancies, adapter templates, and mismatch discovery algorithms, the generation of adapters for coping with a specific incompatibility needs human intrusion and input. Efforts are still needed to offer automatic adapter generation mechanisms that can grasp seamless system integration based on services.

3.1.2 Service adaptation based on formal semantics

To ease automated analysis of service compatibility, business protocols/service behaviors are described by proper semantics such as finite state machine, Petri nets, and process algebra in an amount of studies. The major work in this area divides into three associate-problems: *service compatibility*, *protocol adaptation*, and *behavioral interface adaptation*. In the *service compatibility* category, the focal point is on defining, sensing, and authenticating the compatibilities between services based on proper modeling tools. The *protocol adaptation* aims at generating service adapters based on proper modeling notations. The *behavioral interface adaptation* focus on the problem of fine-tuning behavioral interfaces. Bordeaux et al. [12] explains a service as a labeled transition system, which has a set of states, transitions among states and actions (receive/send messages). Lohmann et al. [45,46] and Martens et al. [47,48] both translate BPEL processes into some sort of Petri nets so that exchanges between two BPEL processes can be evaluated formally and automatically.

Martens et al. spotlights on properly defining and automatically examining compatibility and sameness of service behaviors. Still for behavioral compatibility inspection and verification, Foster et al. [24] alter BPEL processes into a so called finite state process information. Ponge et al. [59] focus on automated analysis of compatibility among Web service protocols with timing constraints.

With the assist of LOTOS process algebra, Mateescu, Poizat, and Salaün [49] affords a series of tools to carry adapter generation. Brogi and Popescu [13] explain targeted BPEL processes into YAWL workflows. Then, an adapter for the YAWL identify workflows is build, which is also in the form of YAWL workflow. The freshly created adapter is examined for deadlocks and installed as BPEL process. Using fuzzy logic, Pernici et al. [57] spotlight on the problem of choosing suitable adaptation approach for QoS changes.

Dumas et al. [20] believes the adaptation of behavioral service interfaces which enclose messages, the order and constraints among these messages. A functional interface is explained by a set of traces over an alphabet made up of communication actions. The authors classify six algebraic transformation operators to arbitrate the dissimilar types of interface mismatches. For behavioral interface adaptation, Quederni et al. [55] model an interface as a figurative transition system. Incompatibilities between user and provider can be computed based on the formal model. From the above discussion, we can study that formal modeling tools/languages offers a powerful means for formally defining and authenticating service compatibilities and also regularly generating adapters for overcoming dissimilarity between Web services/BPEL processes. These approaches need Web services/ BPEL processes to be converted into formal notations in the initial place before any analysis and adaptation can be carried out.

3.2 Process flexibility

In [88], Barbara Weber, proposed 18 change patterns and seven change support features which – in combination – allow for assessing PAIS change frameworks The introduction of change patterns complements existing workflow patterns and permits for more meaningful assessment of existing systems and approaches, particularly if flexibility is an issue In combination with workflow patterns the presented change framework will enable (PA)IS engineers to choose the process management technology which meets their flexibility requirements best.

Zeng et al. [87] sponsors a policy-driven approach for exemption management in BPEL processes. The key plan is the disconnection of the business logic and the exception handling strategy. The specified exception handling strategies are integrated with business logic at runtime to produce exception-aware process schemas.

3.3 Service evolution

Service evolution refers to the process of adjusting a service through a sequence of steady changes [56]. It divides into (i) Web service versioning; (ii) service protocol evolution; and (iii) service evolution. In the sub-problem (i), It spotlights on managing dissimilar versions of WSDL service. In (ii), the importance is on the management for evolving service protocols. In (iii), the spotlight is on providing theorems and methods for managing service evolution.

Service versioning attempts [14,22,23,29,3033,34,41] have been made for enabling Web service versioning, that is, admitting Web services to evolve in a regulation and controlled manner. For example, Kalali et al. [33] put ahead the requirements for developing a Web service registry, called service-oriented monitoring registry, that can observe changes of Web services and reports service requestors when the requested Web services are altered. Kaminski et al. [34] suggest a service design method, called chain of adapters. Here, an adapter that mediates the dissimilarity among the new version and the old version. To ease Web service versioning control, [14,22,28,37] proposes extensions to the latest Web service standards. Brown and Ellis [14] supports the use of version namespace and numbers in UDDI entry to supervise the evolution of Web services. Their method achieves backward compatibility by permitting numerous versions of a Web service to sustain the earlier versions of that service. Diverse to the above work, [23] offers an algorithm called VTracker to notice changes in WSDL documents of successive versions of real-world services, such as Amazon EC2, PayPal SOAP API3 etc. Based on the perceived interface changes, possible effects on the maintainability of service systems are analyzed.

Service protocol evolution. Evolution of service protocols in service based environments is also inspected by a few efforts [66,67,74]. In the framework of service protocols, the problem of fruition is discussed from a static facet (modifying protocol definitions) and a dynamic aspect (managing running protocol cases when the protocol definitions changes especially for long discussions). Ryu et al [66,67] spotlights on dynamic protocol evolution where ongoing discussions (protocol instances) have to to be handled properly when the protocols alters. The researchers also deal with the issue of active conversation exodus when there is no formal protocol for this discussion. A set of methods to examine the impact of a protocol change on the active discussion are devised it also holds forward and backward compatibility. Skogsrud et al. [74] spotlight evolution of confidence negotiation protocols. It finds the varieties of and operators for handling current trust negotiations during a protocol amends are developed.

3.4 Change analysis and Management

Changes in services and business processes will influence each other as a result of the reliance relations between them. In [91], argued about dependencies among services, a change in a assured service can have such an impact on further services that these have to be changed to facilitate keep functioning. An ontological approach to syntactical understanding of service compositions can result in errors in the analysis. A tool that is able to analyze which services are exaggerated when a certain service is adapted (dependencies), how the services are exaggerated when a certain service is adapted (effects), and what is the finest way to deal with this service adaptation (advice). In [89], **Xumin Liu, et.al,** proposed a framework for efficiently managing top–down changes in LCSs, focusing on changes that result in the replacement or addition of a Web service. They proposed a two-phase optimization approach where in the first phase, the selection of Web services is based on using reputation as the key parameter. In the second phase, the non-functional QoWS is used to narrow down the set to those Web services that are both reputable and best meet the QoWS. A formal model to provide the grounding semantics to support the automation of multi change management, including web service ontology and an LCS schema has been proposed. It then defined a set of change operators to specify top-down multi changes based on the formal model. For the change operators, that proposed a set of algorithms to automatically implement them. The correctness of the LCS is also ensured during the process of change enactment. It implemented a prototype of the proposed change enactment process to prove its practicality [92]. With the emerging role of web services in business processes, the requirement of composing and executing them have begun to draw high attention, and today the need to find the optimal web services composition for the business processes is a challenging issue. The proposed technique addresses the issues of conflict detection and check the correctness of the web service composition by selecting an optimal and reliable component web services according to the change requirement. Workflow for the web service composition is generated based on user's request and makes a better web service selection for the composition[1]. The proposed web service selection technique selects a best component web service for the each activity in composition workflow and backup module provides a dynamic backup of a web service by one of its alternative [93]. In [95], The change management process can be made very easily and provides much flexibility. First, the analyst gets the services of the enterprise web service which are available and business logic schema file is loaded for the service logic where he wants to do alterations. After doing changes, the equivalent source code is generated for the changes done in the BL Schema. And finally the web service is re-installed automatically for the changes done in the service code through BL Schema. At the client side, the modified services can be accessed and can be included into the software process. The current system was modeled using FSM (Finite State Machine) for analyzing dependencies where the states are represented as business rules (i.e. functions).

Tudor Dumitras et al. [96] propose a novel framework for distributed implementation of change management by separating the impact assessment (performed by the goal advisors) and the scheduling and business value aggregation (performed by the orchestrator). This approach, centered around a communication protocol rather than on implementation bindings, delegates the appraisal of change-operation impact on service objectives to the objective-manager components in order to leverage their embedded domain-specific knowledge. This also allows us to coordinate changes that span multiple administrative domains and heterogeneous services and software components. The framework takes into account the impact of change management on the enterprise SLOs, the long-term KPI variation and heterogeneous types and sources of change operations (both internal and external). We present three deterministic scheduling algorithms and we compare experimentally the trade-off between their cost and their loss of optimality.

In [97], identified a taxonomy of changes in SOEs using a bottom-up approach. In this approach, describes triggering changes that may occur in Web services. These changes are then mapped to reactive changes in SOEs. A proper change model based on Petri nets to exactly represent these changes has been proposed. It also describes a change management framework based on our change model to provide automatic management of changes in SOEs.

3.4.1 Support for change detection, analysis, and reaction

In [79], change impact analysis has been deliberated from the perception of an organization itself, that is, how changes may crash on its supporting business process. The key idea is to define a set of change impact patterns for capturing diverse types of change consequences. In [94], Change management in service-oriented enterprises has emerged as a new research area entailing frequent nontrivial issues and problems. Impact analysis and change propagation activities in service-oriented environments are more multifaceted and demanding than traditional software systems. This revise was intended to regularly review available literature on change analysis and propagation in service-based BPM systems. As SOA and BPM are becoming leading technologies and their natural coalition is a desirable solution for business agility and strength, noteworthy research work is necessary in this domain. We measured the convergence of SOA and BPM across diverse layers of abstraction. A categorization scheme was proposed in this regard, that covers both horizontal and vertical results of changes across numerous layers. Business processes, policies, process models and complex Web services were considered at the business level, while constituent services and messaging protocols were measured.

Non-functional change management: The study and management for non-functional changes including QoS and strategy changes are not fully examined. Very partial efforts have been prepared for dealing with QoS changes in service compositions. Liu and Bouguettaya [42,43] have recommended some approaches for handling QoS changes in service-oriented ventures. In [11], change impact of QoS assessments on the overall service composition is examined. Impact factor of a constituent service is calculated by considering the dependencies in composition structure and SLAs, to calculate the importance of a Web service in a service composition at runtime. However, still it is hard to understand the collision of non-functional changes in service-based systems. Also, organization support is missing for the higher level of changes in service systems such as policy changes. As a result, classy mechanisms and tools support are insisted for analyzing and responding to these non-functional changes in business collaborations.

4. CONCLUSION

This paper provides an extensive review of the current research on change management in the domain of service-oriented computing. Firstly, we have classified the major types of changes that may happen to services into two broad categories: functional change and non-functional change. Then, we have reviewed the main research on managing these types of changes from four broad categories: service adaption, service evolution and change analysis, and management in service compositions. In each sector of research, we have discussed the major approaches proposed in the literature to coping with various types of change problems and pointed out critical issues that remain to be solved. More efforts are required for managing non-functional changes including QoS changes and higher level of change: policy-induced changes in the context of inter-organization collaboration.

References

1. Adams M, ter Hofstede AHM, Edmond D, van der Aalst WMP (2006) Worklets: a service-oriented implementation of dynamic flexibility in workflows. In: Proceedings of OTM conferences (1). Montpellier, France, pp 291–308
2. Agarwal V, Jalote P (2010) From specification to adaptation: an integrated QoS-driven approach for dynamic adaptation of web service compositions. In: Proceedings of the 2010 IEEE interna-tional conference on web services (ICWS 2010). Miami, Florida, USA, pp 275–282
3. Akram MS, Bouguettaya A (2004) Managing changes to virtual enterprises on the semantic web. In: Proceedings of the 5th international conference on web information systems engineering (WISE). Brisbane, Australia, pp 472–478
4. Akram MS, Medjahed B, Bouguettaya A (2003) Supporting dynamic changes in web service environments. In: Proceedings of 1st international conference on service oriented computing. Trento, Italy, pp 319–334
5. Andrikopoulos V, Benbernou S, Papazoglou MP (2008) Manag-ing the evolution of service specifications. In: Proceedings of the 19th international conference on advanced information systems engineering (CAiSE). Montpellier, France, pp 359–374
6. Andrikopoulos V, Benbernou S, Papazoglou MP (2009) Evolving service from a contractual perspective. In: Proceedings of the 20th international conference on advanced information systems engi-neering (CAiSE), Amsterdam, the Netherlands, pp 290–304
7. Banerjee J, Kim W, Kim HJ, Korth HF (1987) Semantics and imple-mentation of schema evolution in object-oriented databases. In: Proceedings of the 1987 annual conference on association for com-puting machinery special interest group on management of data. California, San Francisco, pp 311–322
8. Benatallah B, Casati F, Grigori D, Motahari-Nezhad HR, Toumani F (2005) Developing adapters for web services integration. In: Pro-ceedings of the 17th international conference on advance informa-tion systems engineering (CAiSE). Porto, Portugal, pp 415–429
9. Benatallah B, Casati F, Toumani F (2006) Representing, analysing and managing web service protocols. Data Knowl Eng 58:327–357
10. Berardi D, Calvanese D, Giacomo GD, Lenzerini M, Mecella M (2005) Automatic service composition based on behavioral descriptions. Int J Coop Inf Syst 14(4):333–376
11. Bodenstaff L, Wombacher A, Reichert M, Jaeger MC (2009) Analyzing impact factors on composite services. In: Proceedings of the 2009 IEEE international conference on services computing. Bangalore, India, pp 218–226
12. Bordeaux L, Salaün G, Berardi D, Mecella M (2004) When are two web services compatible? In: Proceedings of the 5th interna-tional workshop on technologies for e-services. Canada, Toronto, pp 15–28
13. Brogi A, Popescu R (2006) Automated generation of BPEL adapters. In: Proceedings of the 4th international conference service-oriented computing. Chicago, IL, USA, pp 27–39
14. Brown K, Ellis M (2004) Best practices for web services version-ing. IBM Technical library. ttp://www.ibm.com/ developerworks/ webservices/library/ws-version/
15. Casati F, Ceri S, Pernici B, Pozzi G (1998) Workflow evolution. Data Knowl Eng 24:211–238.
16. Charfi A, Mezini M (2007) AO4BPEL: an aspect-oriented exten-sion to BPEL. In: World wide web, pp 309–344
17. Choi J, Cho Y, Shin K, Choi J (2007) A context-aware workflow system for dynamic service adaptation. In: Proceedings of the 2007 international conference on computational science and its applica-tions. Kuala Lumpur, Malaysia, pp 335–345
18. de Alfaro L, Henzinger TA (2001) Interface automata. In: ESEC/SIGSOFT FSE. pp 109–120
19. Dorn C, Dustdar S (2010) Interaction-driven self-adaptation of ser-vice ensembles. In: Proceedings of the 22nd international confer-ence on advanced information systems engineering. Hammamet, Tunisia, pp 393–408
20. Dumas M, Spork M, Wang K (2006) Adapt or perish: Algebra and visual notation for service interface adaptation. In: Proceedings of the 4th international conference on business process management. Austria, Vienna, pp 65–80
21. Evans H, Dickman P (1997) Drastic: a run-time architecture for evolving, distributed, persistent systems. In: Proceedings of the 11th European conference on object-oriented programming. Springer, Finland, pp 243–275
22. Fang R (2007) A version-aware approach for web service direc-tory. In: Proceedings of the IEEE international conference on web services (ICWS 2007), pp 406–413

23. Fokaefs M, Mikhaiel R, Tsantalis N, Stroulia E (2011) An empirical study on web service evolution. In: Proceedings of the 2011 IEEE international conference on web services (ICWS), pp. 49–56

24. Foster H, Uchitel S, Magee J, Kramer J (2004) Compatibility verification for web service choreography. In: Proceedings of the 2004 IEEE international conference on web services. San Diego, California, pp 738–741

25. Gerede cE, Hull R, Ibarra OH, Su J (2004) Automated composition of e-services: lookaheads. In: Proceedings of the 2nd international conference on Service oriented computing. ACM, New York, NY, USA, pp 252–262

26. Gong S, Xiong J, Liu Z, Zhang C (2010) Correcting interaction mis-matches for business processes. In: Proceedings of the 2010 IEEE international conference on services computing. Miami, Florida, USA, pp 457–465

27. Grossniklaus M, Leone S, de Spindler A, Norrie MC (2010) Dynamic metamodel extension modules to support adaptive data management. In: Proceedings of the 22nd international confer-ence on advanced information systems engineering. Hammamet, Tunisia, pp 363–377

28. Hallerbach A, Bauer T, Reichert M (2008) Managing process vari-ants in the process life cycle. In: Proceedings of the 10th inter-national conference on enterprise information systems. Barcelona, Spain, pp 154–161

29. Harney J, Doshi P (2006) Adaptive web processes using value of changed information. In: Proceedings of the 4th international conference on service-oriented, computing. pp 179–190

30. Harney J, Doshi P (2007) Speeding up adaptation of web ser-vice compositions using expiration times. In: Proceedings of the 16th international conference on world. Banff, Alberta, Canada,1023–1032

31. Jarouchech Z, Liu X, Smith S (2010) Apto: a MDD-based generic framework for context-aware deeply adaptive service-based processes. In: Proceedings of the 2010 IEEE internatinal conference on web services. Miami, Florida, USA, pp 219–226

32. Joeris G, Herzog O (1998) Managing evolving workflow specifi-cations. In: Proceedings of the 3rd IFCIS international conference on cooperative information systems. New York City, New York, USA, pp 310–319

33. Kalali B, Alencar P, Cowan D (2003) A service-oriented monitor-ing registry. In: Proceedings of the 2003 conference of the centre for advanced studies on collaborative research (CASCON ’03), pp 107-121

34. Kaminski P, Mller H, Litoiu M (2006) A design for adaptvie web service evolution. In: Proceedings of the 2006 international workshop on self-adaptation and self-managing systems. Shanghai, China, pp 86–92

35. Kataoka Y, Ernst MD, Griswold WG, Notkin D (2001) Automated support for program refactoring using invariants. In: Proceedings of the international conference on software maintenance. pp 736–743

36. Kongdenfha W, Saint-Paul R, Benatallah B, Casati F (2006) An aspect-oriented framework for service adaptation. In: Proceedings of the 2nd international conference on service-oriented computing (ICSOC). USA, New York, pp 15–26

37. Koning M, ai Sun C, Sinnema M, Avgeriou P (2009) VxBPEL: supporting variability for web services in BPEL. Inf Softw Technol 51(2):258–269

38. Kramer J, Magee J (1990) The evolving philosophers prob-lem: dynamic change management. IEEE Trans Softw Eng 16(11):1293–1306

39. Lanza M, Ducasse S (2002) Understanding software evolution using a combination of software visualization and software metrics. In: Proceedings of L’OBJET vol 8, no (1–2), pp 135–149

40. Lehman MM (1984) Program evolution. Inf Process Manag 20(1):19–36

41. Leitner P, Michlmayr A, Rosenberg F, Dustdar S (2008) End-to-end versioning support for web services. In: Proceedings of the IEEE international conference on services computing (SCC ’08).

42. Liu X, Bouguettaya A (2007) Managing top-down changes in service-oriented enterprises. In: Proceedings of the 2007 IEEE international conference on web services. Salt Lake City, Utah, USA, pp 1072–1079

43. Liu X, Liu C, Rege M, Bouguettaya A (2010) Semantic support for adaptive long term composed services. In: Proceedings of the 2010 IEEE international conference on web services (ICWS 2010). Miami, Florida, USA, pp 267–274

44. Liu Y, Ngu AHH, Zeng L (2004) QoS Computation and Policing in Dynamic Web Service Selection. In: Proceedings of the WWW 2004. USA, New York, pp 66–73

45. Lohmann N, Massuthe P, Stahl C, Weinberg D (2006) Analyzing interacting BPEL processes. In: Proceedings of the forth interna-tional conference on business process management (BPM 2006),

46. Lohmann N, Massuthe P, Stahl C, Weinberg D (2008) Analyzing interacting WS-BPEL processes using flexible model generation. Data Knowl Eng 64(1):38–54

47. Martens A (2005) Analyzing web service based business processes. In: Proceedings of the 8th international conference fundamental approaches to software engineering. Edinburgh, UK, pp 19–33

48. Martens A, Moser S, Gerhardt A, Funk K (2006) Analyzing compatibility of bpel processes. In: Proceedings of the advanced international conference on telecommunications and international conference on internet and web applications and services. Guade-loupe, French Caribbean, p 147

49. Mateescu R, Poizat P, Salaün G (2008) Adaptation of service pro-tocols using process algebra and on-the-fly reduction techniques. In: Proceedings of the 6th international conference on service-oriented computing (ICSOC 2008). Australia, Sydney, pp 84–99

50. Mens T, Tourwe T (2004) A survey of software refactoring. IEEE Trans Softw Eng 30(2):126–139

51. Mietzner R, Leymann F (2008) Generation of BPEL customiza-tion processes for saas applications from variability descriptors. In: Proceedings of 2008 IEEE international conference on services computing. Honolulu, Hawaii, US, pp 359–366

52. Motahari-Nezhad HR, Benatallah B, Martens A, Curbera F, Casati F (2007) Semi-automated adaptation of service interactions. In: Proceedings of the 16th international conference on world wide web. Banff, Alberta, Canada, pp 993–1002

53. Motahari-Nezhad HR, Xu GY, Benatallah B (2010) Protocol-aware matching of web service interfaces for adapter development. In: Proceedings of the 19th international conference on world wide web. Raleigh, North Carolina, USA, pp 731–740

54. Nitto ED, Ghezzi C, Metzger A, Papazoglou MP, Pohl K (2008) A journey to highly dynamic, self-adaptive service-based applica-tions. Autom Softw Eng 15(3–4):313–341

55. Ouederni M, Salaün G, Pimentel E (2011) Client update: A solu-tion for service evolution. In: Proceedings of the 2011 IEEE inter-national conference on services computing (SCC). pp 394–401

56. Papazoglou MP (2008) The challenges of service evolu-tion. In: Proceedings of the 20th international conference advanced information systems engineering. France, Montpellier,pp 1-15

57. Pernici B, Siadat SH (2011) A fuzzy service adaptation based on qos satisfaction. In: Proceedings of the 23rd international conference advanced information, systems engineering, pp 48-61

58. Pesic M, Schonenberg MH, Sidorova N, van der Aalst WMP (2007) Constraint-based workflow models: change made easy. In: Proceedings of OTM conferences (1). Vilamoura, Portugal, pp 77-94

59. Ponge J, Benatallah B, Casati F, Toumani F (2007) Fine-grained compatibility and replaceability analysis of timed web service protocols. In: Proceedings of the 26th international conference on conceptual modeling (ER 2007). Auckland, New Zealand, pp 599-614

60. Ponnekanti S, Fox A (2004) Interoperability among inde-pendently evolving web services. In: Proceedings of the 2004 ACM/IFIP/USENIX international middleware conference. Canada, Toronto, pp 331–351

61. Rajlich V (1997) A model for change propagation based on graph rewriting. In: Proceedings of international conference on software maintenance. Bari, Italy, pp 84–91

62. Reichert M, Dadam P (1998) Adept$_{flex}$-supporting dynamic changes of workflows without losing control. J Intell Inf Syst 10(2):93–129

63. Reichert M, Rinderle S, Dadam P (2003) On the common support of workflow type and instance changes under correctness constraints. In: Proceedings of the 2003 OTM confederated international con-ferences, CoopIS, DOA, and ODBASE. Catania, Sicily, Italy, pp 407-425.

64. Rinderle S, Weber B, Reichert M, Wild W (2005) Integrat-ing process learning and process evolution—a semantics based approach. In: Proceedings of the 3rd international conference (BPM 2005). Nancy, France, pp 252–267

65. Rinderle S, Wombacher A, Reichert M (2006) Evolution of process choreographies in dychor. In: Proceedings of the 2006 OTM con-federated international conferences on CoopIS, DOA, GADA, and ODBASE. France, Montpellier, pp 273–290

66. Ryu SH, Casati F, Skogsrud H, Benatallah B, Saint-Paul R (2008) Supporting the dynamic evolution of web service protocols in service-oriented architectures. ACM Trans Web 2(2), Article 13

67. Ryu SH, Saint-Paul R, Benatallah B, Casati F (2007) A frame-work for managing the evolution of business protocols in web services. In: Proceedings of the 4th Asia-Pacific conference on con-ceptual modelling (APCCM2007). Ballarat, Victoria, Australia, pp 49-95

68. Sadiq SW (2000) Handling dynamic schema change in process models. In: Proceedings of the 2000 Australasian database confer-ence, pp 120–126

69. Sadiq SW, Orlowska ME, Sadiq W (2005) Specification and val-idation of process constraints for flexible workflows. Inf Syst 30(5):349–378

70. Seguel R, Eshuis R, Grefen PWPJ (2010) Generating minimal pro-tocol adaptors for loosely coupled services. In: Proceedings of IEEE international conference on web services. Miami, Florida, USA, pp 417–424

71. Serral E, Valderas P, Pelechano V (2010) Supporting runtime sys-tem evolution to adapt to user behaviour. In: Proceedings of the 22nd international conference on advanced information systems engineering. Hammamet, Tunisia, pp 378–392

72. Shan Z, Kumar A, Grefen PWPJ (2010) Towards integrated service adaptation. In: Proceedings of the IEEE international conference on web services. Miami, Florida, USA, pp 385–392

73. Skarra AH, Zdonik SB (1986) The management of changing types in an object-oriented database. In: Proceedings of the 1986 con-ference on object-oriented programming systems, languages, and applications, pp 483–495

74. Skogsrud H, Benatallah B, Casati F, Toumani F (2007) Man-aging impacts of security protocol changes in service-oriented applications. In: Proceedings of the 29th international conference on software engineering (ICSE 2007). Minneapolis, MN, USA, 468–477

75. Sun H, Zhao W, Yang J, Su J (2011) Ticobtx-net: a model to manage temporal consistency of service oriented business collaboration. IEEE Trans Serv Comput 5(2):207–219

76. Taentzer G, Goedicke M, Meyer T (1998) Dynamic change man-agement by distributed graph transformation: towards configurable distributed systems. In: Proceedings of the 6th international work-shop on theory and application of graph transformations. Pader-born, Germany, pp 179–193

77. van der Aalst WMP, Basten T (2002) Inheritance of workflows: an approach to tackling problems related to change. Theor Comput Sci 270(1–2):125–203

78. van der Aalst WMP, Weske M, Grünbauer D (2005) Case handling: new paradigm for business process support. Data Knowl Eng 53(2):129–162

79. Wang Y, Yang J, Zhao W (2012) Change impact analysis in service-based business processes. Serv Oriented Comput Appl 6(2):131–149

80. Weber B, Reichert MR, Rinderle-Ma S (2008) Change patterns and change support features - enhancing flexibility in process- aware information systems. Data Knowl Eng 66(3):438–466

81. Weidlich M, Weske M, Mendling J (2009) Change propagation in process models using behavioural profiles. In: Proceedings of the 2009 IEEE international conference on services computing. Bangalore, India, pp 33–40

82. Wombacher A (2009) Alignment of choreography changes in BPEL processes. In: Proceedings of the international conference on services computing (SCC). Bangalore, India, pp 1–8

83. Wombacher A, Fankhauser P, Neuhold EJ (2004) Transforming BPEL into annotated deterministic finite state automata for service discovery. In: Proceedings of the IEEE international conference on web services. San Diego, California, USA, pp 316–323.

84. Wu Y, Doshi P (2007) Regret-based decentralized adaptation of web processes with coordination constraints. Proceedings of 2007 IEEE international conference on services computing. Salt Lake City, Utah, USA, pp 262–269

85. Yamashita M, Becker K, Galante R (2011) Service evolution man-agement based on usage profile. In: Proceedings of the 2011 IEEE international conference on web services (ICWS). pp 746–747

86. Yu C, Popa L (2005) Semantic adaptation of schema mappings when schemas evovle. In: Proceedings of the 31st VLDB confer-ence. Trondheim, Norway, pp 1006–1017

87. Zeng L, Lei H, Jeng JJ, Chung JY, Benatallah B (2005) Policy-driven exception-management for composite web services. In: Pro-ceedings of the 7th IEEE international conference on e-commerce technology. München, Germany, pp 355–363.

88. Barbara Weber, Manfred Reichert, Stefanie Rinderle-Ma (2008) Change patterns and change support features – Enhancing flexibility in process-aware information systems, Data and knowledge Engineering, pp 438-466.

89. Xumin Liu, Athman Bouguettaya, Qi Yu· Zaki Malik (2001), Efficient change management in long-term composed services, SOCA(2001), 5:87-103.

90. Alex Norta, Paul Grefen, Nanjangud C. Narendra (2014),' A reference architecture for managing dynamic inter-organizational business processes, Data and knowledge Engineering 91, pp 52–89.

91. Linda Terlouw (2012), Analyzing the Evolution of Web Services using Fine-Grained Changes, TUD-SERG-2012.

92. B. Muruganantham K. Vivekanandan P Kirubanantham (2014), Dynamic Web Services Composition Using Change Management System, International Journal of Engineering Development and Research.

93. M.Monicavinodhini, C.Kanimozhi (2014) Enhancing Change Management in Long Term Composed Services, International Journal of InnovativeResearch in Science, Engineering and Technology.

94. Khubaib Amjad Alam et. al (2015), Impact analysis and change propagation in service - oriented enterprises: A systematic review Information systems pp 43–73

95. Thirumaran. M, Dhavachelvan. P and Naga Venkata Kiran. G (2012), A Collaborative Framework For Managing Run-Time Changes In Enterprise Web Services International Journal of Web & Semantic Technology (IJWesT) DOI: 10.5121/ijwest.2012.3306 85.

96. Tudor Dumitras, Daniela Rosu, Asit Dan, Priya Narasimhan, (2006)Dynamic Change Management for Minimal Impact on Dependability and Performance in Autonomic Service-oriented Architectures. CMU- CyLab -06-003 CyLab, Carnegie Mellon.

97. Salman Akram Athman Bouguettaya (2010), A Change Management Frame work for Service Oriented Enterprises International Journal of Next-Generation Computing (IJNGC), Vol. 1, No. 1, 09 2010.