Towards A Framework for Supporting Dynamic Evolution of Services in the GUISET Infrastructure

Sandile W. Dlamini, Paul Tarwireyi and Mathew O. Adigun
Department of Computer Science
University of Zululand, Private Bag X1001, KwaDlangezwa, Empangeni 3886
Tel: +27 35 902 6393, Fax: +27 35 902 6569
email: sandilewilmoth@gmail.com, {ptarwire, madigun}@pan.uzulu.ac.za

Abstract - In service-oriented architectural based systems, it is very rare that services can remain unchanged; services in such environments have to constantly evolve in order to meet the ever changing consumer requirements. However this evolution needs to happen in a controlled manner, as unregulated changes can cause severe disruptions to service consumers and may require radical modifications. Hence, there is a need to develop mechanisms that will ensure the continued use and availability of services in the event that changes have been made to them. Managing and supporting service evolution in Service-Oriented Architecture (SOA) is very complicated and is still a topic of ongoing research. GUISET (Grid-based Utility Infrastructure for SMME Enabling Technology) is one of the SOA-based infrastructures, in which controlling service changes has not been addressed. This work proposes a framework to ensure that service evolution is supported, controlled and managed effectively in such a way that any service change that occurs in the GUISET infrastructure is not disruptive or does not alter the way in which service consumers conduct their business.

Index Terms – SOA, Service Evolution, GUISET

I. INTRODUCTION

Service-Oriented Architecture (SOA) is an architectural style that has recently became one of the preferred choices as a way of designing, developing, deploying and managing systems that are characterized by coarse-grained services and service consumers [1]. In service-oriented environments, services have to constantly evolve in order to meet the different and ever changing consumer requirements. Service evolution may result in service consumers being disrupted, thus it needs to be properly managed. According to [2], supporting and managing service evolution in SOA is very complicated and is still a topic of ongoing research. Service evolution is defined as the continuous process of developing a service through a series of consistent and unambiguous changes [2], which may come from the modification of existing functionalities in a service, an introduction of a new functionality to a service and/or introducing new policy constraints. Service changes as described in [3] can be distinguished into two types; shallow changes and deep changes. These changes can occur at any stage in the service lifecycle and to identify and understand their impact on service consumers is a very critical and challenging issue. However according to [4], the impact can be described and/or defined in terms of compatibility.

The Grid-based Utility Infrastructure for SMME Enabling Technology (GUISET) proposed by [5], is a SOA-based infrastructure that enables services to be discovered, composed, and invoked in order to support the business processes of Small, Medium and Micro Enterprises (SMMEs). In GUISET, services are shared among consumers with different and varying requirements, thus they cannot remain static or unchanged. GUISET services have to constantly evolve in order to meet the needs and requirements of consumers who want services to change as their business changes. However, service evolution in GUISET has never been addressed, thus this work aims to produce a framework that will ensure that service evolution is supported and managed in the GUISET infrastructure. The evolution problem relates to many research disciplines, particularly to Software Configuration Management (SCM) and Component-Based Systems (CBS). SCM is a discipline of controlling the evolution of large and complex software systems [6] and has contributed in major ways to software evolution and maintenance. SCM systems were initially used for managing critical software and they are currently used to manage the evolution of any kind of software [7]. But this (SCM) technology cannot be directly applied and/or fall-short in supporting large and distributed environments like GUISET, due to the fact that SCM tends to focus on the file artifact, ignoring higher levels of abstraction and assume a centralized control with respect to the evolution of software artifacts [8]. In CBS, works such as [9] and [10] builds a type system for components and defines a set of conditions by which the components can evolve compatibly. These works are in principle geared towards dealing with changes; however their theories cannot be adopted unchanged in the context of service evolution.

In addressing the issue of service evolution in the GUISET infrastructure, inspiration and valuable techniques can be drawn from these (SCM and CBS) disciplines and other related ones. The rest of this paper is organized as follows: Section II discusses related work; our proposed research is discussed in Section III. Section IV concludes the paper.

II. RELATED WORK

Existing works in the field of service evolution can be split into two groups. A group of adaption-based approaches such as [11] and [12], which focuses at implementing adapters in order to work out service-client mismatches. These (adaption-based) approaches raise a number of issues with respect to service evolution. One of the issues is that, adaption does not occur in reaction to change; it may be the

actual cause of change. Another issue with adaption-based approaches is that they do not incorporate compatibility checks before generating suitable adapters; this is one of the key missing steps in these approaches. Thus according to [13] adaption is one means by which evolution manifest. The other group of existing works [14] and [15] allow designers to change service interfaces in order to ensure compatibility. The techniques used in these works, in modifying service interfaces can be divided into two main steps; they start by detecting service changes through calculating the differences between the new and an old version of a service and then propagate those changes into the new service version. These approaches uses models that do not take into consideration message parameters as well as internal behaviors, also their compatibility check does not allow one to detect the changes to be handled. Therefore, drawing inspiration from these works, our solution to GUISET will also be based on a compatibility analysis method. However we intend to provide a method that will automatically detect the changes to be handled.

III. PROPOSED RESEARCH

The main goal of this work is to develop a service compatibility and versioning mechanism for solving the GUISET service evolution problem of controlling service changes. We intend to ensure that consumers who want stability in services and consumers who want services to change; are both supported and satisfied. The key objectives of this research work are:

i. To investigate and draw GUISET service evolution requirements through the creation of usage scenarios.
ii. To formulate a typical GUISET service evolution model.
iii. To develop and prototype a framework for experimental evaluation of the model formulated in objective (ii).

IV. CONCLUSION

The development of tools, mechanisms and methods to support, manage and govern service evolution in service-oriented environments is still at the early stage. This work proposes a framework to support the evolution of services in GUISET, an infrastructure that leverages on service-oriented, grid, and utility computing to enable services to be shared among SMMEs. The aim is to enhance GUISET and add value to the ongoing research on service evolution.

REFERENCES

[1] G. Lewis, D. B. Smith, and K. Kontogiannis, "A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems," Carnegie Mellon University, Software Engineering Institute, 2010.

[2] M. P. Papazoglou, "The Challenges of Service Evolution," in *20th International Conference on Advanced Information Systems Engineering (CAiSE 2008*, 2008, pp. 1–15.

[3] V. Andrikopoulos, S. Benbernou, and M. P. Papazoglou, "On The Evolution of Services," *IEEE Transactions on Software Engineering*, vol. (in pre-print, to appear in 2012), 2012.

[4] R. Fang, L. Lam, L. Fong, D. Frank, C. Vignola, Y. Chen, and N. Du, "A Version-aware Approach for Web Service Directory," in *Web Services, 2007. ICWS 2007. IEEE International Conference on*, 2007, pp. 406 –413.

[5] M. Adigun, O. Emuoyibofarhe, and S. Migira, "Challenges to Access and Opportunity to use SMME enabling Technologies in Africa," Johannesburg South Africa, 2006.

[6] W. F. Tichy, "Tools for Software Configuration Management," in *International Workshop on Software Version and Configuration Control*, Grassau, Germany, 1988, pp. 1–20.

[7] J. Estublier, D. Leblang, A. van der Hoek, R. Conradi, G. Clemm, W. Tichy, and D. Wiborg-Weber, "Impact of Software Engineering Research on the Practice of Software Configuration Management," *ACM Transactions on Software and. Engineering Methodology.*, vol. 14, no. 4, pp. 383–430, Oct. 2005.

[8] S. Sowrirajan and A. van der Hoek, "Managing the Evolution of Distributed and Interrelated Components," in *Proceedings of the 2001 ICSE Workshops on SCM 2001, and SCM 2003 Conference on Software Configuration Management*, Berlin, Heidelberg, 2003, pp. 217–230.

[9] P. Brada, "Component Revision Identification Based on IDL/ADL Component Specification," in *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, New York, NY, USA, 2001, pp. 297–298.

[10] M. Zenger, "Type-Safe Prototype-Based Component Evolution," in *Proceedings of the 16th European Conference on Object-Oriented Programming*, London, UK, 2002, pp. 470–497.

[11] S. Becker, A. Brogi, I. Gorton, S. Overhage, A. Romanovsky, and M. Tivoli, "Towards an Engineering Approach to Component Adaptation," in *SPRINGER-VERLANG, LNCS*, 2006, p. 2006.

[12] J. Camara, J. A. Martin, G. Salaun, J. Cubo, M. Ouederni, C. Canal, and E. Pimentel, "ITACA: An Integrated Toolbox for the Automatic Composition and Adaptation of Web Services," in *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*, 2009, pp. 627 –630.

[13] G. Canfora, "Software Evolution in the Era of Software Services," in *Proceedings of the Principles of Software Evolution, 7th International Workshop*, Washington, DC, USA, 2004, pp. 9–18.

[14] V. Andrikopoulos, S. Benbernou, and M. P. Papazoglou, "Evolving Services from a Contractual Perspective," in *Proceedings of the 21st International Conference on Advanced Information Systems Engineering*, Berlin, Heidelberg, 2009, pp. 290–304.

[15] F. Casati and B. Benatallah, "Supporting the Dynamic Evolution of Web Service Protocols in Service-Oriented Architectures." *ACM Trans. Web*, vol. 2,2008.

BIOGRAPHY

Sandile W. Dlamini is a first year Masters student at the University of Zululand, with interest in Service Evolution.