Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 6 (2012) 163 – 170

Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

Evaluation of Change Factors for Web Service Change Management

Thirumaran.M[a] , Dhavachelvan.P[b], Aishwarya.D[c], Kiran Kumar Reddy[d]

[a]*Department of Computer science and Engineering, Pondicherry Engg College, India*
. [b]*Department of Computer science and Engineering, Pondicherry University, India.*
[c]*Department of Computer science and Engineering, Pondicherry Engg College, India.*
[d]*Department of Computer science and Engineering, Pondicherry Engg College, India.*

Abstract

Service oriented architecture (SOA) is a smart designing principle which has been evolved for integrating business tasks. Business activities that have to be designed based on SOA are implemented via web services. Using web services (WS) one can exchange data between different applications and different platforms. Service providers register their services in the service registry and consumer obtains the required services from the same. The main concern in this routine which directly sways business growth rate is change management. Change management is an emerging issue in web service computing where clients might want to change the obtained services at some period of time. But in order to do it they should be requesting the provider programmers each and every time and separate payment has to be done for that task. In order to reduce this complexity we propose a new model for implementing change requests by business analysts themselves. Here we propose a new dynamic schema driven business logic model using Finite State Machine (FSM) to accomplish WS change management in a best manner so that business growth rate can be increased. This model is distinctively done for business analysts to perform changes in the services on their own instead of depending on the programmers. Furthermore a predictive model is contrived using cellular automata for supporting business analysts. The predictive model includes the change factors like order of execution; similarity measure, schema validation, and mapping function and time/space complexity which appears when a particular change request is executed.

Keywords: Web Services; Change Management; Change factors; Finite State Machine; Cellular Automata.

1. Introduction

SOA can be viewed in terms of various levels of abstractions, ranging from those that concern within an application to those that concern service applications interacting across enterprises. In this architecture,

doi:10.1016/j.protcy.2012.10.020

change management plays a big issue. It completely affects the business growth due to the vast competition that is present between enterprises based on customer satisfaction. According to the changes in the real world or changes in the need of the customers, the web services provided also should be changed and provided. These changes can be made slowly or immediately. The change that has to be made immediately affects the business growth tremendously because the changes will cause sudden change in the usage of the service. According to the change made, the customers might start using it more or less and accordingly our profit will get increased or lowered. We can consider ticket reservation as example, sometimes the ticket fare can be lowered or increased and according to the trend the end customers might start using our clients' service. But every time when the fare amount has to be changed, the business analyst cannot contact the provider programmer which will introduce a delay in the change management process. He might feel more comfortable when he himself is provided with an option to change the services provided to him. So we introduce a business logic model that allows the business analyst himself to make the required changes according to his own customers' needs. Here we use business logic model instead of business process model because here the process is not implemented from the scratch, only small changes has to be made to the existing source codes. The changes can be made at three levels – rule, function and parameter. The changes can also be extended to dependency between two rules or functions or parameters. Since the business analyst himself is allowed to make the changes the language which is used here to represent the code involved for a web service is XML. The machine process able schema gives a clear hierarchical business workflow. The statements will be available which can be directly understood and the changes can be accomplished. This paper further says about the literature survey made in section 2, describes the new proposed idea with algorithms for change measures in section 3 and case study made on banking domain in section 4 with conclusion and references in section 5 and 6 respectively.

2. Literature survey

Yang et al presented a method for detecting failures in implementation of services that are running currently in concurrent order and also from rules in Service Oriented architecture based atmospheres. It also finds the dependencies that co exist among the services and are extracted from log files when any change occurs [1]. However this approach has various shortcomings such as no graphical views are generated for the dependencies existing among the various service logics and fails for correctness of the solution. We use tools to generate a graph for the existing dependencies when a rule in the service logic has to be modified. Business dealings are obligatory to accomplishing the company of changes within legislation and strategy. However the successful executions of innovative legislation are a lot expensive, might include long escort period which cause huge loss to industry trade. Yiwei Gong and others proposed a novel technique for business processes in favor of building litheness and agility while applying changes through innovative or improvised policies. It also elaborated a new research method for software reuse by joint policy execution [2]. But this technique fails to bring agility and flexibility in business logic as compared to business process. Dimitris Apostolouet all presented an ontology driven method for managing changes in e-Government services based on SOA. It enables methodical reaction of these systems to changes by applying proper techniques intended for accomplishing stability whenever a change is exposed [3]. However this method deals at business process level and not at business logic layer. In [4], the authors presented a novel method to represent changes in ontology which have been organized in layers and provided data independence from versions, etc. Also deposits of change circulation stratagems are allowed to keep distributed copy of the same ontology synchronized. This method does not clearly indicate how the changes in ontology are mapped at business layers. A methodology for handling changes in Long-term Composed Services (LCS) was proposed by Xumin Liu, Athman Bouguettaya and others. In this technique the authors have categorized changes in two ways:- Top-down and Bottom-Up scaffolds. Also the functional and non-functional properties of the composed services have been identified [5], [6],[13] and [14]. When compared with the work described in this paper, business logic is taken as importance as business analysts prefer implementing the business logic directly and not interested in hierarchy of business process. We also have identified the QoS parameters into account as it has to satisfy both