Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 4 (2013) 99 – 109

Procedia IERI

www.elsevier.com/locate/procedia

2013 International Conference on Agricultural and Natural Resources Engineering

Conventional Usage of Finite State Machine over Petri net InWeb Service Change Management Framework

Thirumaran.M[a] , Dhavachelvan.P[b], Aishwarya.D[c], K.Rajakumarid

[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

Finite state machine and Petri nets are one among the various conceptual and computational models that have been widely used in analyzing diverse web service research areas. In this paper a comparison of using finite state machine and Petri nets for modeling a web service change management framework is analyzed with experimental analysis. Though there are many existing change management approaches to satisfy the normal changes, they fail to there is no proper support for efficient change evaluation and monitoring. Also complex structure called Petri nets is used to model the overall system functionality when a change whereas simple finite state machine can be used for the same representation since the process of change management only includes emergency and minor changes. Using FSM, only a specific logic can be extracted instead of exacting the overall function as in case of Petri nets. Thus for the evaluation of changes implemented, FSM is very efficient for the representing the overall change management. In this paper, we are going to see how FSM is efficient in modeling the change management and also the evaluation of changes implemented using certain change factors when compared to Petri nets.

Keywords: Web Services; Change Mangement;Change factors;Finite State Machine;Petri nets.

1. Introduction

Change management is a set of processes that are employed to ensure that significant changes are implemented to a business process during its maintenance phase. The purpose of the change management process is to ensure that: Business risk is managed and minimized; Standardized methods and procedures are used for efficient and prompt handling of all changes; all changes to service assets and configuration items are recorded in the configuration management system; and all authorized changes support business needs and goals. Hence there arises the need for an effective framework managing those changes without affecting the business functionality. Here Change Management framework is proposed for making minor alterations to business logic but whose effect is more pronounced to the profitability of the organization. So the changes in business logic of the web service must be done very quickly within the given time constraint as the services consumed by the providers and clients must not be affected there by managing the changes at the runtime. We focus on creating environment for dynamic variations to the business logic so that it is feasible to make frequent modifications in a service which aids service provider and serves him to satisfy clients' newfangled quests. Such environment is useful in the run-

doi:10.1016/j.ieri.2013.11.016

time management of web services and to exactly spot the solution to the service provider's maintenance element. There are many models that subsist for Business Process Management in which the process is recycled for the overall process changes. However, the problem with these solutions is that they only support the process level flexibility and not the application/service level flexibility. On the other hand this Business Logic Management framework tries to append that service level flexibility. These changes are done at the Business analyst level instead of being done at the Developer level, which reduces the hierarchy level in change management and thus implies a reduction in time and cost requirements. Here comes the need for an efficient model representation which provides clear visualization of the process of change management and the evaluation of changes implemented for the analyst to understand the system easily. Petri nets provide a graphical notation for the formal description of the dynamic behavior of systems and are particularly well suited to systems which exhibit concurrency, synchronization, mutual exclusion and conflict whereas finite state machines is a mathematical model of computation used to design both computer programs and sequential logic circuits. Though Petri nets have been used in the application areas such as Software design, Workflow management, Process Modeling, Data analysis, Concurrent programming, Reliability engineering, Diagnosis, Discrete process control and Simulation, finite state machine plays an efficient role in representing the evaluation of change implemented. Also Petri net is not only a significant burden placed on the analyst in order to specify complex models, but in addition the graphical representation may become too complex to be useful and understandable. In order to provide simple understandable and useful representation for the inexperienced non specialist analyst, finite state machine can be used for implementing the framework. This paper fully concentrates on the advantages of using FSM over Petri nets for change management and evaluation of changes implemented.

2. Literature survey

Xumin Liu et al. [1] proposed an Ev-LCS, an end-to-end framework that specifies, reacts to, and verifies top-down changes in a LCS. This framework first propose a formal model which provides the grounding semantics to support the automation of change management and a set of change operators that allow specifying a change in a precise and formal manner by proposing a set of algorithms to automatically implement them. It then proposes a change enactment strategy that actually implements the changes. Dimitris Apostolou et al. [2] proposed an ontology-based approach for developing and maintaining e-Government services that can effectively deal with changes which enables the systematic response of e-Government systems to changes by applying formal methods for achieving consistency when a change is discovered and also enables the knowledgeable response of service designers and implementers to changes by utilizing design rationale knowledge. Florian Rosenberg et al. [3] proposed a domain-specific service selection mechanism and system implementation to address the issue of runtime adaptation of composite services that implement mission-critical business processes by making use of a domain-specific language called VieDASSL which can be used by domain experts to define the runtime adaptable selectors based on the QoS attributes in the adaptive QoS model. This approach ensures that changes in the QoS model and selectors can be handled at runtime without the disruption of the business processes by assuming the units of measurement for each QoS attribute are fixed and cannot be dynamically adapted. Bassam Atieh Rajabi and Sai Peck Lee [4] proposed two formalism of BPEL namely Graph based and Rule based formalism. The rigidity in graph based models incurs problems lack of runtime criteria (flexibility, dynamic and adaptability, which compromises the ability of the graph based processes to react to dynamic changes in BP and exceptional circumstances whereas the common objective of BRMS is to integrate complex process logic into a process model to support dynamic changes. To increase flexibility in process execution, Constraint based BP management approach is used which supports the evolution of BP. In [5], An adaptive and flexible framework is proposed to integrate OO diagramming technique and PN modeling language (i.e. the graph based which has the visual appeal of being intuitive and explicit, even for those who have little or no technical background and rule based modeling language which requires good understanding of propositional logic and the syntax of logical expressions) in order to increase the representation capability for graph based modeling to support the dynamic changes in the runtime instances. In order to generate the BPEL code automatically, Yanhuna Du et al [6] proposed the Petri Net approach that automatically can verify the composition of partially incompatible services and also serves the time in such a way which will generate the BPEL code. This approach consist of 3 phases: Modeling composition of services is modeled as open WorkFlow Nets (oWFNs) which are composed by using the mediation transaction (MTs). Automatic verification of composition uses graph called Modular reachability graph (MRG) of composition is constructed and analyses the compatibility that can significantly improve the problem of state space explosion. In order to save the time, techniques called Event-condition Action (ECA) rule based is