Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 889 – 894

Procedia
Computer Science

The 2nd International Workshop on the Adaptation of Web Services (AWS 2013)

Evolution Feature Oriented Model Driven Product Line Engineering Approach for Synergistic and Dynamic Service Evolution in Clouds:Four Kinds of Schema

Zhe Wang, Kevin Chalmers

School of Computing
Edinburgh Napier University
EH10 5DT, Edinburgh, UK

Email:{z.wang2, k.chalmers}@napier.ac.uk

Abstract

The proposed research will focus on developing a novel approach to solve Software Service Evolution problems in Computing Clouds. The approach will support dynamic evolution of the software service in clouds via a set of discovered evolution patterns. An initial survey informed us that such an approach does not exist yet and is in urgent need. Evolution Requirement can be classified into evolution features; researchers can describe the whole requirement by using evolution feature typology, the typology will define the relation and dependency between each features. After the evolution feature typology has been constructed, evolution model will be created to make the evolution more specific. Aspect oriented approach can be used for enhance evolution feature-model modularity. Aspect template code generation technique will be used for model transformation in the end. Product Line Engineering contains all the essential components for driving the whole evolution process.

KeyWords: Evolution Feature, Evolution Pattern; Clouds Computing; Service Evolution; Model Driven Product Line Engineering;

Corresponding author . Tel .: +44 0131 455 2303 +44 07747464018
Email Address: z.wang2@napier.ac.uk (Zhe Wang) ,k.chalmers@napier.ac.uk(Kevin Chalmers)

doi:10.1016/j.procs.2013.06.120

1. Introduction

What is Evolution Feature?

- Evolution feature is the modeling of evolution requirement come from people interaction.
- Evolution feature is the collection of evolution task.
- Evolution features have common aspect and unique aspect.
- Evolution feature can be divided and assembled.
- Evolution feature model is a tree.
- Nodes in the tree have dependency from its parents.
- The typology of the model is the formal description of the evolution task, each node is an individual task.
- Evolution process sequence and dependency can be described by the evolution feature tree.

What is Evolution Pattern?

- Evolution pattern can be defined by two parts, the static part and the dynamic part.
- The static part describes the details of the evolution while the dynamic part describes the process of the evolution.
- Both of them are extendable and adaptable based on the changing requirements and the evolution problems.
- Several static parts can be composed together to generate a new dynamic process or be divided into smaller unit and make each as an independent execution process.
- Class and state diagram can be used to describe the static and dynamic parts respectively.

Evolution pattern can be defined by two parts, the static part and the dynamic part. The static part describes the details of the evolution while the dynamic part describes the process of the evolution. Both of them are extendable and adaptable based on the changing requirements and the evolution problems. Several static parts can be composed together to generate a new dynamic process or be divided into smaller unit and make each as an independent execution process. Class and state diagram can be used to describe the static and dynamic parts respectively.

Evolution feature and evolution pattern can be combined to solve one particular problem, the evolution feature describes how to solve the problem while the evolution pattern solves the problem in real situation, and evolution feature is actually the formal description of the problem solving methodology. The evolution patterns will each corresponding with a problem domain and solution domain. By compare the evolution feature and evolution pattern, developer can select the appropriate pattern[1][2][3].

2 The Approach and Realization

As the service in clouds is large in scale and complex in structure so that its evolution will be extremely different compare with local or server side software evolution. In order to make the evolution in clouds more effective and efficiency the evolution pattern will be categorised into server groups, each of which can solve one kind of evolution problem. As shows in Figure.1 there are five groups of evolution pattern proposed in this approach. Those are service dependency dealing, SaaS structure transformation, SaaS evolvability enhancement, SaaS system dynamic enhancement and SaaS function enhancement. As shows in Figure.7 the evolution mechanism will be designed as a corresponding supporting architecture to drive the pattern into execution.

In order to make the evolution process more reliable and fault tolerant , map the evolution process into several parallel process by using map reduce will reduce the risk of evolution collapse. Evolution threads is the carrier of each evolution process unit, it can be extended without suspend the whole evolution process.

Evolution Requirement can be classified into evolution features; researchers can describe the whole requirement by using evolution feature typology, the typology will define the relation and dependency between each features. After the evolution feature typology has been constructed, evolution model will be created to make the evolution more specific. Aspect oriented approach can be used for enhance evolution feature-model modularity. Aspect template code generation technique will be used for model transformation in the end. Product Line Engineering contains all the essential components for driving the whole evolution process.

The whole approach will be realized based upon the most popular clouds computing framework named Hadoop. As shows in Figure.9 it is mainly composed of three parts, evolution pattern repository, evolution mechanism and evolution interaction framework. The process of the evolution will be driven by its mechanism under the workflow defined by the pattern, as evolution pattern itself will be transformed into workflow in the execution level. Pattern configurations will be defined by the evolution interaction framework. Each pattern has its own corresponding evolution mechanism in order to realize its specific feature. The evolution mechanism can be designed as an extend platform that can meet the requirements for kinds of evolution variants, its common aspect can be reused by all kinds of patterns. The interaction framework is the corridor for the interaction among developer, pattern and mechanism, working as a controller and monitor in the whole evolution process. The details of pattern will also be defined by using the interaction framework[1]. Figure 1 shows the approach framework[1][2][3].

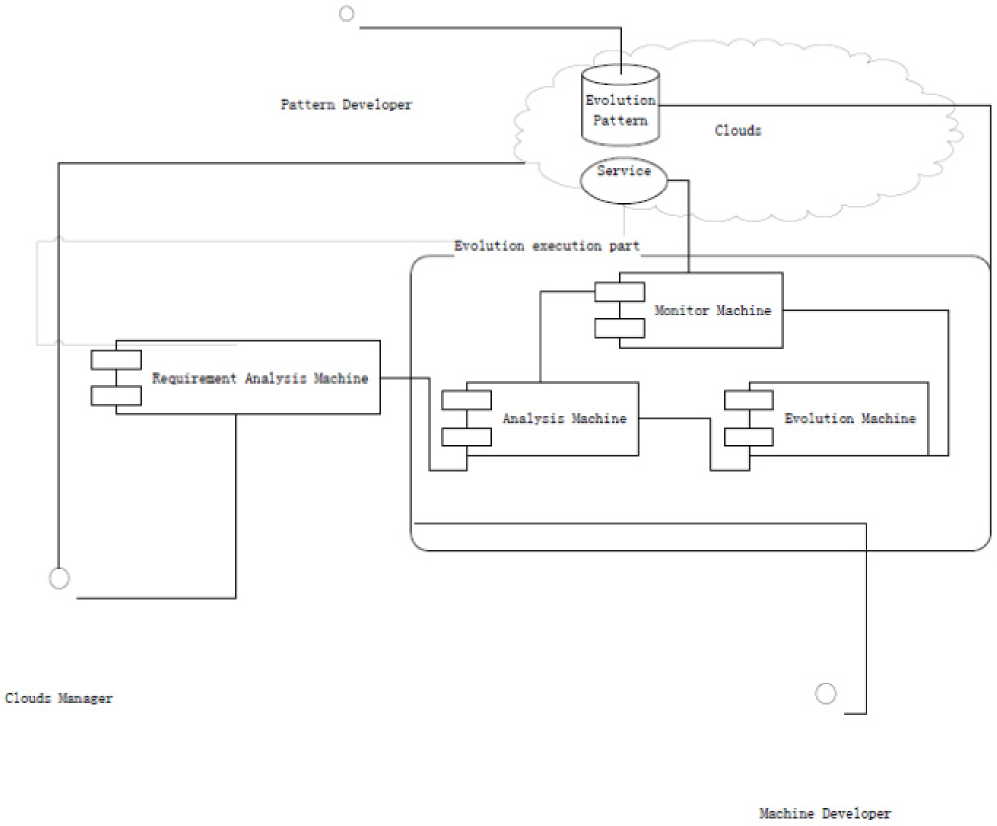

Figure 1. The whole approach framework

3 Four Kinds of Schema

Data Structure can also be called as meta-data or schema for defining the meat-element we need before it to be largely expanded and instantiated. In the approach there are four types of data structure need to be defined. These structures will be associated together in order to generate a connected evolution process. Feature , Pattern , Weaving and Repository will take its own position by related association in the data structure definition respectively . Structure itself is also a kind of data. Its higher than specific data, it can be used to define the structure of the specific data. With out clearly define the structure , we can not make the specific data to be organized in a unified way cause we need to make the approach achieved systematically .

Feature Oriented Evolution Requirement Description Data Structure (Synergistic+Clouds Feature)

Feature Oriented Data Structure will define how the feature structured, it will classify the evolution requirement proposed by people and problem it going to solve. It can be expanded in order to adapt with the changing environment and problem itself. The association with Evolution Pattern Data Structure will make the selection of the appropriate pattern can be happened automatically. In this project, we will set the rang before define the feature's data structure, which is never to exceed the proposed ten problems. The feature's construction process is the realization of the aim of being Synergistically evolution cause the feature comes from interaction, it takes from the people's evolution requirement.

Evolution Pattern Description Data Structure(Problem+Clouds Feature)

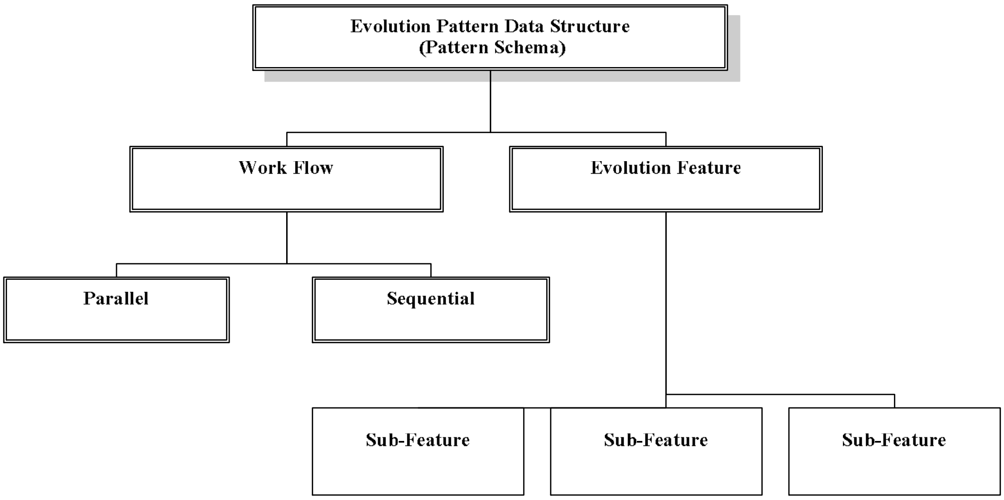

Figure 2 Evolution Pattern Data Structure

The Evolution Pattern Data Structure will define the skeleton of the essential elements that will affect the aim of the evolution. It will contain the work flow part and evolution function part. The work flow part exactly define the working manner of the evolution function, for instance, executed sequentially or parallelly, even under conditional decision before execution.The evolution function has its associated relation with evolution feature that will make the pattern function selection by the feature model can be done automatically, pattern model and the feature model has its one-one correspondence relation. The work flow and pattern function will be packaged together in order to generate several evolution aspects as the entity for execution. Inside the aspects , the evolution work flow has already define the working manner for the evolution function, sequentially, parallelly or conditionally.Figure 2 shows the Evolution Pattern Data Structure.

Aspects Weaving Description Data Structure(Dynamic+Clouds Feature)

The weaving data structure is used to define the weaving process in a unified format. The weaving process will deeply affect the dynamic feature of the service evolution process. In order to make the evolution process as seamless as possible ,optimization on the weaving mechanism is urgently needed. The aspects weaving data structure is such a meta-data that used to define how the whole weaving process to be realized. Several aspects has been generated from the **Evolution Pattern Data Structure** and now we need to weaved it into the Original Service Execution Process. When and how to weave is a question that qualified to be answered. The answered of the question will deeply affect the seamless requirement of the evolution process , however, it will be perfectly solved by the **Aspects Weaving Data Structure Definition.**

Pattern Description Data Structure in Repository (Reuse+Clouds Feature)

Pattern stored inside the repository need an unified way of description , except whatever what kinds of problem it can solved and what kinds of situation it used to tackle. These patterns need a specific kind of data structure to encapsulate it , like components be to be wrapped and then exposed as web service. Evolution pattern also need to be wrapped before stored into the repository. The pattern stored should has an association relation with pattern itself in order to make it could be find automatically based on pattern inquiring[1].

4 Conclusion

The current state-of-the-art evolution methods are not enough for Clouds Service evolution, especially for SaaS, it lacking the run-time resolvability to support the challenging features of clouds, such as dynamism to suit constantly changing needs in business, close collaboration among all stakeholders, virtualization, trustworthiness and efficiency. Traditional software evolution methods are not capable in this case. New approaches and tools are highly in demand for cloud service evolution. We propose to engage a novel synergistic and pattern-driven approach to improve dynamic cloud service evolution in a heuristic manner with healthiness validated. An initial framework has been proposed to tackle above identified problem.

5 Acknowledgment

The work in this paper has been jointly sponsored by the British Royal Society of Edinburgh (RSE-Napier E4161) and the Natural Science Foundation of China (Ref: 61070030).

Special thanks for Dr Kevin Chalmers ,Lu Yanjing and my parents for their kindness encouragement on me.

6 Reference

[1] Wang, Z., Xiaodong,L., ,Chalmers, K.,Cheng, G.EvolutionPattern for Service Evolutionin Clouds.In: The7th International Conference for Internet Technology and Secured Transactions (ICITST-2012). London, UK: Copyright © ICITST-2012 PublishedbyInfonomicsSociety,UK.,2012

[2] Wang,M.,Bandara,K.,Pahl,C..Process as a Service - Distributed Multi-tenant Policy-based Process Runtime Governance.

2010 IEEE International Conference on Services Computing 978-0-7695-4126-6/10,Dublin City University,2010

[3]Yuan,P., Jin,H.,Yuan,S.,Cao,W., Jiang,L. WFTXB: A Tool for Translating Between XPDL and BPEL.The 10th IEEE International Conference on High Performance Computing and Communications.978-0-7695-3352-0/08 $25.00 © 2008 IEEE.Huazhong University of Scien ce and Technology, Wuhan, 430074, China ,2008