Evolution Feature Oriented Model Driven Product Line Engineering Approach for Synergistic and Dynamic Service Evolution in Clouds

Zhe Wang, Xiaodong Liu, and Kevin Chalmers
School of Computing, Edinburgh Napier UniversityEH10 5DT, Edinburgh, UK
Email:(z.wang2, x.liu, k.chalmers)@napier.ac.uk

Guojian Cheng and Hongchun Wei
School of Computer Science, Xi'an Shiyou (Petroleum) University, DianZi 2nd Road 18, Xi'an, China
Email:(gjcheng, hcwei)@xsyu.edu.cn

***Abstract*— 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.**

***Index Terms*—evolution feature, evolution pattern; clouds computing; service evolution; model driven product line engineering**

I. INTRODUCTION

A. What is Software Evolution?

Software evolution is part of software engineering, it will make the software can continuously capture the new requirement from people, including function requirement and non-function requirement. Function requirement is to make the system can do more for people whether non-function requirement is mainly related to quality of software , software architecture optimization , system evolvability , system stability , system robust, etc.

B. Why We Need Service Evolution in Clouds?

In order to achieve the software evolution in clouds environment, we need specific approach for service evolution in clouds, service here means software as service for people. Service can be published through web service or EJB to the clouds. People using this service through invoke it interface by knowing its WSDL description. Service can be organized together through BPEL; BPEL is flexible in order to adapting itself with changing requirement and environment in order to get the best system performance. It seems to be that service evolution in clouds is just evolution of these web service and BPEL which re-organized them together, but the fact is that we must take the clouds feature into consideration in order to maximize the advantage we can get to support service running in clouds. Such feature will certainly affect the service evolution in clouds cause service will be branded with clouds feature after it has been evolved by our approach. We will both consider function and non-function requirement in the approach [1] [2] [3].

II. THE APPROACH

A. How to Achieve the Approach?

The software evolution approach will have six major steps in the Evolution Life Cycle which are Evolution Requirement Specification, Evolution Design Definition, Evolution Implementation, Evolution Execution, Evolution Testing and Further Evolution.

B. Evolution Requirement Specification

The evolution requirement specification is the vital part of the evolution life cycle. The quality we have for the evolution requirement specification will directly affect the quality of service evolution in clouds. In order to make the service evolution specification we get from people is more accurate, flexible and reusable a new description mythology is introduced which is Evolution Feature Oriented Requirement Specification. By using that method we can get more accurate requirement specification details from interaction with people. Further process will happen by using these features with a feasible combination, which will generate an optimal evolution plan. Without the plan we can not get to the design phrase, however not to mention the implementation. In this phrase people including service user, service evolution analysis, service and clouds

Manuscript received December 19, 2012; revised February 30, 2013.

doi: 10.12720/jiii.1.1.6-9

manager will be involved. Use case model and Evolution feature model will be used to modeling the plan.

C. Evolution Design Definition

The evolution design includes choosing the appropriate evolution pattern from the pattern repository to complete and meet the requirement of the evolution plan. That will be done by the service evolution designer, the service designer will choose the pattern corresponding with the evolution feature constructed by the evolution analysis , service user, service and clouds manager in the first phrase. Use case model, class diagrams, state diagrams, interaction diagrams will used to describe the whole picture in different aspect of view. Evolution pattern itself will take advantage of the clouds feature into its own design [4].

D. Evolution Implementation

The evolution implementation will be driven by the model driven production line engineering from design to code. Design Models we get in the second phrase will be used in this phrase. All the designs including evolution pattern itself will be transformed into code at this time. Actually evolution pattern has already been integrated into the whole design picture in the second phrase; it has been absorbed as a part of the design, especially the vital part of the whole design. In this phrase we can only focus on code realization for the perfect design has been done in the last phrase [4].

E. Evolution Execution

The evolution execution will be driven by the evolution mechanism in order to achieve the aim people proposed in the first phrase. The evolution mechanism will support the evolution pattern respectively for its working purpose is different. The common part of the mechanism will support the common evolution process without special or unique demands because the whole the evolution pattern is a part of the whole implementation, there must be common aspect exist in the execution. Hadoop will be the fundamental component for the mechanism design [4].

F. Evolution Testing

Evolution testing will mainly focus on finding system bug and collect information which will guidance its further evolution in the future. The testing will involved system robust, system reliability, system evolvability , system performance, etc.

G. Further Evolution

Further evolution will happen cause requirement and system environment will keep changing all the time.

H. What is Evolution Feature?

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.

- 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.

I. Pattern Meta-Model Example

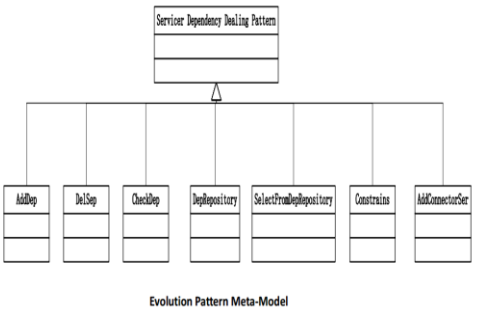

Figure 1. Evolution pattern Meta-model

As shows in Fig. 1. The pattern meta-model is the description of the aim that the evolution process will achieve, it is the content the evolution task should complete in the process. The meta-model will be driven by the evolution aspect generation model in order to reach the execution level by using pattern aspect template. The execution will be guidance by the evolution process model which describe how to use the pattern aspect template, it can be thought as the dynamic description of the evolution pattern while meat-model is the static description of the evolution pattern [4].

J. The Evolution Aspect Generation Model

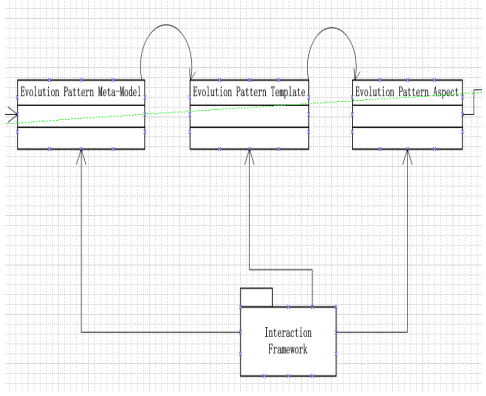

Figure 2. Synergistic pattern aspect generating model

As shows in Fig. 2. The evolution aspect generation model will used the iteration process model as the fundamental generation technique. The iteration process will be happened under the guidance of interaction framework.

K. Process Model

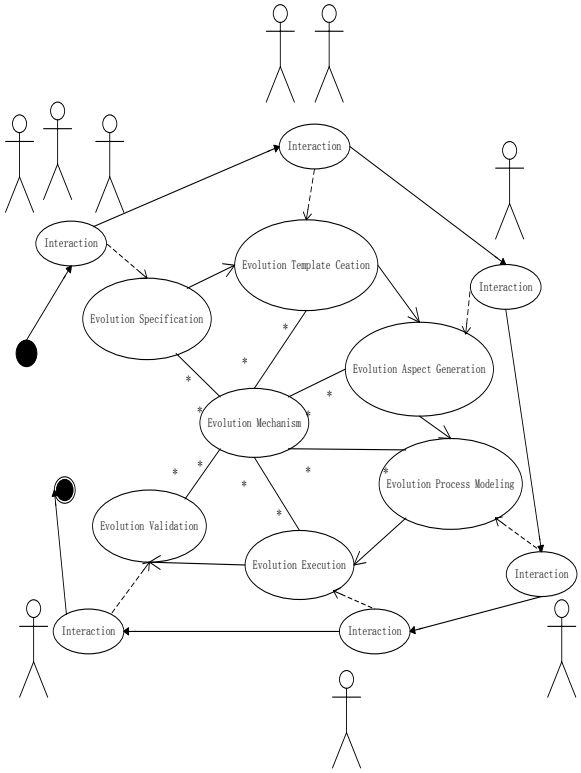

Figure 3. Evolution Process Model

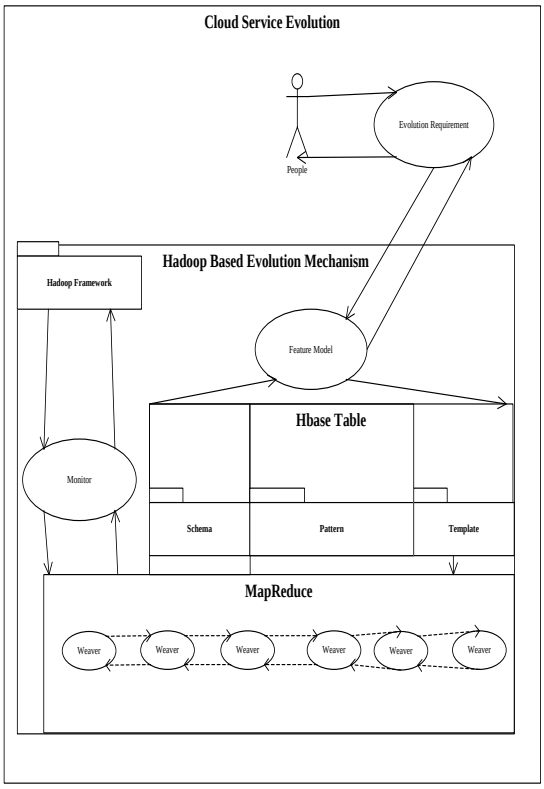

Figure 4. Evolution mechanism design model.

As shows in Fig. 3 and Fig. 4. The whole evolution process will be guidance under the spiral and iteration software development model. People will involve into the process by interaction with evolution framework in order to define the specific evolution requirements. That will make the evolution can real meet the people's demand. The whole process can be categorized into evolution specification, evolution template creation, Evolution aspect modeling, evolution execution and evolution verification.

L. Evolution Mechanism

Weaving mechanism is part of the evolution mechanism.Waving is the last step of the evolution execution process, it will directly affect the time consuming ,dynamism and flexibility of the evolution mechanism.Hadoop Map and Reduce Methodology will be introduced into Mechanism design.

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.

- Scalable
- Lightweight
- Multiple workflows
- Map and Reduce
- Workflow description language
- Workflow storage
- Decentralized workflow execution management
- Task scheduling
- Fault tolerance
- Dynamic

III. Related Work

A. Unified Approach for the Dynamic Evolution of Context-aware Services

Service in a context-aware environment need dynamic evolution urgently for its user requirement continuously changing quickly, this paper propose an approach for the development of evolution of context-aware service, It adopts MDD (Model-Driven Development) to enable services specification in a high-level modeling language and automatic run-time generation of their executable implementations, hence contributes significantly to both design and evolution flexibility and cost savings. But the reuse of the evolution does not effectively achieved by the approach ,each evolution process in the approach can be encapsulated as an evolution pattern by using the pattern approach to driven the whole process will greatly improve the current approach to a more reusable pattern based evolution[5].

IV. Conclusion

Hadoop cluster can be used to construct a small empirical research environment as Platform as Service. The evolution mechanism based on that can be thought as Software as Service. Hadoop cluster can control and manage the platform in order to make affect on the

evolution process, both of them has corresponding activity feature to each other. Evolution Pattern can guidance how these corresponding activity feature happened. Pattern will be the corridor between Hadoop cluster and evolution mechanism. Pattern Schema will have its specification on Hadoop cluster management, evolution mechanism working management and pattern itself [4].

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. Guojian Cheng, Dr Xiaodong Liu, Dr Kevin Chalmers and Prof. Hongchun Wei for their kindness encouragement on me.

REFERENCES

[1] Z. Jaroucheh, X. Liu, and S. Smith, "An Approach to Domain-based Scalable Context Management Architecture in Pervasive Environments," *Personal and Ubiquitous Computing*, vol. 16, no. 6, pp. 741-755, 2012.

[2] H. Yang and X. Liu, *Software Reuse in the Emerging Cloud Computing Era*, Pennsylvania, USA: IGI Global Publishing, 2012.

[3] Z. Jaroucheh, X. Liu, S. Smith, and H. Zhao, "Lightweight software product line based privacy protection scheme for pervasive applications," in *Proc. 35th IEEE COMPSAC. Munich, Germany: IEEE Computer Society*, 2011.

[4] Z. Wang, X. D. Liu, K. Chalmers, and G. Cheng, "Evolution pattern for service evolution in clouds," in *7th International Conference for Internet Technology and Secured Transactions*, London, UK: Published by Infonomics Society, UK, 2012.

[5] Z. Jaroucheh, X. Liu, S. Smith, "A unified approach for the dynamic evolution of context-aware services," in *Proc. International Conference on Innovations in Computers, Information and Communication*, India: PSG Tech, 2012.

Wangzhe is a Phd research student in Edinburgh Napier University , his work is finding an innovative approach for Service evolution in Clouds. He has published the paper Evolution Pattern for Service Evolution in Clouds. In: The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012). London, UK: Copyright © ICITST-2012 Published by Infonomics Society, UK.

Dr Xiaodong Liu is an active researcher in software engineering with known reputation and leading expertise in context-aware adaptive services, service evolution, mobile clouds, pervasive computing, software reuse, andcomponent-based systems in Edinburgh Napier University.

Dr Kevin is an researcher in software engineering in Edinburgh Napier University, his work is mainly focus on computer graphics , algorithms and emergent computing model.

Dr Cheng Guojian is a professor in Xi'an Shiyou (Petroleum) University, has published 30 paper indexed by the (SCI/ISTP/EI) library and he has published one research book on *Incremental Learning of Self-Organizing Variable Topology Neural Networks*. His work is mainly focus on artificial intelligence, software architecture and digital oil field.

Wei Hongchun is a professor in Xi'an Shiyou(Petroleum) University. He has published 16 papers; his work is mainly focus on software engineering and information system.