Adaptive Human Management System Architecture
based on Reengineering for MDA and CBD

Haeng-Kon Kim
School of Information Technology, Catholic University of Daegu, Korea
hangkon@cu.ac.kr

Abstract. In this paper, we suggest AHMS(Adaptive Human Management Systems) architecture based reengineering approach using design patterns. Design pattern, as core element of software architecture, has integrated the concept of standardization about certain domain and expert experience into a set of related component that can perform specific functionality with better structure. We describe a reengineering process that defines target architecture to be transformed by refining architecture information of legacy system extracted through domain analysis, identifies the reengineering patterns that are applicable in that architecture, and completes target system by mapping identified reengineering pattern into target architecture.

Keyword: Mobile Cloud Computing (MCC), cloud computing

1 Introduction

A legacy system is an application that was developed on older technology and is past its prime use, but is still critical to current businesses[1]. Therefore, this legacy system is being viewed as an asset that represents an investment that grows in value rather than a liability whose value depreciated over time[2]. Most legacy systems are being pressured to concurrently respond to escalating requirements. That is, legacy systems are faced with changing industry models such as e-Business, globalization, changing business models such as time-to-market, CRM, emerging information technologies such as Internet, Open system, and new platforms, and emerging new information architectures such as J2EE, Web service and component reuse[3]. Especially, to reduce complexity and cost, and to allow easier change management and exploitation of emerging new requirements, we have recognized that WWW(World Wide Web) and component technology as mechanism of improved information integration, must be accepted. Because these technologies allow that we can access various information across distributed resources by providing the implementation of the application is transparent to the service offered via interfaces.

In this paper, we have intention to use design pattern for reflecting specific knowledge within domain of legacy system to elements of target system, and utilizing
as the solutions suggested by preceded experts about common problems that must solve in reengineering process.

2 Related Works

There is CORUMII (Common Object-based Reengineering Unified Model II)[4] defined in SEI CMU as reengineering methodology to be referred most widely. CORUM model is reengineering tool interoperation to include software architecture concepts and tools. The extended framework - called CORUMII- is organized around the metaphor of a “horseshoe”, where the left-hand side of the horseshoe consists of fact extraction from an existing system, the right hand side consists of development activities, and the bridge between the sides consists of a set of transformations from the old to the new.

As another reengineering methodology, there is MORALE (Mission ORiented Architecture Legacy Evolution)[5] that developed in Georgia Institute of Technology. It features an inquiry-based approach to eliciting change requirements, a reverse engineering technique for extracting architectural information from exist code, an approach to impact assessment that determines the extent to which the existing system’s architectural components can be reused in the evolved version, and a specific technique for dealing with the difficulties that arise when evolving user interface.

Especially, the similar approach to our paper including pattern-based reengineering is in [6]. But, these researches must accept information damage that can happen while maps design decision information to codes because they only emphasize the analysis of legacy codes. Also, because is very hard to identify pattern structure from code, their researches are considering extraction of structural pattern of Gamma, and is dependent to specific legacy language. Recently, research that aim at pattern identification of all types(structural, creation, and behavior patterns of Gamma) through UML-based meta-modeling, is progressing[4].

3 Adaptive Human Management System Architecture based on Reengineering

3.1 Reengineering process

This paper first composes architecture of target system with design information of legacy system that is recovered through understanding of various reference resources including domain analysis. And, by identifying and mapping reengineering design
patterns that is set of component that embody this architecture, complete architecture-based reengineering process. So, overall reengineering process suggested in this paper is summarized in (Figure 1).

3.2 Transformation servlet programs into EJB components

With grasped domain information and reference architecture, architecture of target system transforming servlet programs to EJB component is like a (figure 2). Top part in figure has the roles that analyze servlet program and identify core modules in architecture of reengineering system. In middle part, extract business logics of EJB component with analysis information, create transformation information for EJB environment, and generate components so that may fit in EJB specification. In last part, creates interface form that exhibit access interface of whole system to user.

Pattern-based architecture for AHMS modeled by mapping patterns is suitable to
target architecture MDA and CBD. Abstract Factory and Template pattern are applied to acquire family information related to domain and generate various type of legacy information. Template pattern used to analyze business information of various language included in servlet and stores these information as transparent access way. Also, Composite and Facade pattern are used for processing to transform various types in unit form and for providing unified interface. And Strategy pattern is used to prepare generation rules variously according to EJB component types and to apply these rules as independent algorithm in generating real codes. Target system for Servlet2EJB could be supported the guidelines for design decision of individuation patterns offered in boundary of pattern structure. Also, could develop classes of target system by structure that is coincided in pattern.

4 Conclusion

In this paper, we describe a reengineering process that defines target architecture of legacy system by refining architecture information extracted through domain analysis, identifies the reengineering patterns that are applicable in that architecture, and completes target system by mapping identified reengineering pattern into target architecture. Also, we explain the Servlet2EJB project transforming servlet programs into EJB components as a case study for realizing our reengineering process. Servlet2EJB is prototyping system analyzing servlet programs, extracting component information and generating EJB components. Now, we have suffered from collecting reengineering patterns and library them. Also, we are in difficulties for lack of systematical procedures and guideline to evolve our approach. In future, we will plan to establish formal algorithm for extracting reengineering patterns from legacy source code and related documentations. Also, we will construct supporting tool including repository to manage components, patterns and architecture for reengineering process.

Acknowledgement. "This work (Grants No.C0124408) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013"

References


519 Copyright © 2013 SERSC
4. SEI Reengineering Center "Perspectives on Legacy System Reengineering, 1995
5. Rick Kazman, Steven G. Woods, S. Jeromy Carriere, “Requirements for Integrating
   Software Architecture and Reengineering Models: CORUM II”, Fifth Working Conference