2012 4th International Conference on Computer Research and Development
IPCSIT vol.39 (2012) © (2012) IACSIT Press, Singapore

Software Evolution based on Service-Oriented Requirement in Internetware

Yuhong Jia[1a], Shang Zheng[2b]

[1]Information Engineering College, JiLin Business and Technology College, Chang Chun, 130062, China
[2]Software Technology Research Laboratory, De Montfort University, Leicester, LE1 9BH, UK
[a]376116940@qq.com, [b]szheng@dmu.ac.uk

Abstract. Internet techniques have been changing tremendously due to globalising computing environment. Open, dynamic, uncertain and uncontrollable features of internet environment have brought new challenges and complexity to traditional software technologies and software running platform. In order to resolve these new challenges, Internetware has been proposed as an emerging and new software paradigm, which should be autonomous, cooperative, evolvable, and trustworthy. In this paper, the status and explorations of Internetware paradigm will be discussed and analysed. According to the features of Internetware paradigm, a service-oriented requirements approach of reengineering techniques on how to make software application evolvable in Internet computing environment is proposed in the paper. Meanwhile, an example shows how to use the proposed approach and framework is illustrated. The paper also discusses the ongoing research and future research of Internetware design.

Keywords: component; formatting; style; styling; insert

1 Introduction

Due to the rapid development of computer hardware and software, the demands and costs of software changes are increasing continuously. How to perform software evolution [1] is now widely recognized as a crucial aspect of building enterprise application.

The rapid development of Internet techniques has make the Internet Computing become an emerging mainstream computing platform, on which new versions of software systems are developed, deployed and executed. The large and different amount of entities and services has been seen through the various approaches of deploying and integrating in the global computing environment. For example, Grid Computing [2], Service Computing [3], Ubiquitous Computing [4] and Cloud Computing [5, 6].Specifically, in Cloud Computing, software will be treated as service (Software as a Service) that runs on top of a virtualised environment of resource pool. Actually, any of these services and paradigms cannot execute without the Internet infrastructure. In these years, the fast development of the Internet has made it become a globally ubiquitous infrastructure. Besides its basic connecting function, the Internet is also growing up into a computer that has the ability of global and ubiquitous. Now, it not only consists of a tremendous number of distributed computing devices, but also serves as a platform with much more computing resources for solving more problems that are complicated.

However, during the popularity of the Internet, a wide range of Internet applications has placed new demands and challenges such as openness, portability to adapt quickly reconfiguration. Meanwhile, in the Internet Computing environment, more and more goals are changing, uncontrollable or unclear before completing finally. In other words, the open, dynamic and distributed natures of the Internet Computing have decided that the traditional software model and software engineering approaches cannot be suitable for the emerging computing environment. With these changes, many traditional software applications have become legacy ones because traditional software model or software engineering approaches can just make software

systems run in a few restricted individual computers, single function and be limited to a small group of developers. In order to adapt the natures of Internet computing environment and save enterprise cost through reusing the legacy software systems, the software paradigm must be more and more autonomous, evolvable, situational and coordinative. For the emerging features of software systems on the Internet, a new software paradigm for the Internet computing environment need to be designed and developed, which can support the development, deployment and provide good quality for software applications on the Interment.

According to the new requirement for software paradigm, researchers have proposed the Internetware [7, 8, 9], which means software paradigm for the Internet. Many researchers from Chinese universities and institutes, for example, Peking University, Nanjing University, Tsinghua University, and IBM China Research Lab and so on have participated in this project. The Internetware project studies, establishes, and applies quantifiable methods for software development and deployment in the emerging computing environment.

This paper will present the concept and the features, and introduce current progress of Internetware research first. However, the Internetware needs to keep itself updating with new requirements of computing environment and end users. In order to ensure software system evolution in Internetware, the paper proposes how to utilise service requirements analysis approach of software reengineering techniques to make sure that the structure and function of the Internet entities can meet the dynamic change in the Internet computing environment. The paper is organised as follows. Section II describes the concept and features of the Internetware paradigm, and states key challenges for developing Internetware application. Section III presents the traditional types of techniques of software reengineering, and proposes a service-oriented requirement approach to analyse and ensure Internetware applications can update and evolve while the new service requirement comes. Section IV illustrates a simple example on how to use the proposed framework and approach, and finally in Section V discusses the future research work and ends this paper with conclusion.

2 Internetware

With the transformation to the open, dynamic and distributed computing environment, the traditional software paradigms have become more unsuitable in environment setting and coordination operation and so on. There already exist some related observations and proposals in this field. For example, Open Resource Coalitions [10], Cyber-Physical Systems [11] and so on. However, a new software paradigm that can be more suitable for these new requirements in open and dynamic computing environment is still needed.

Therefore, researchers in China proposed the Internetware that is constructed by a set of autonomous and distributed software entities over the Internet, together with a set of various coordination connectors enabling the operations among these entities in the Internet computing environment.

As shown in Fig. 1 [12], the Internetware software entities collaborate with other ones on demand. Meanwhile, the se entities can also self-organise the application domain or community of interest. According to the research in China, the development of Internetware can be recognised as continuous and various distributed resources into the ordered software systems.

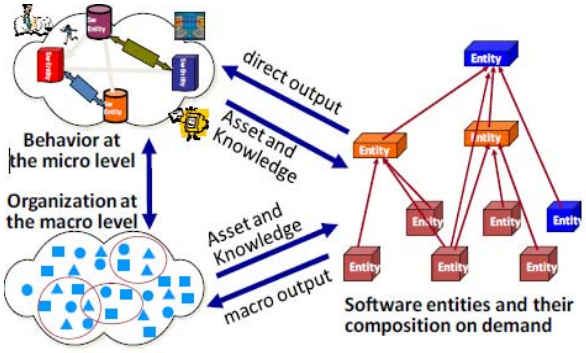

Figure 1. Behaviour, Organisation and Composition Modes of Internetware

Distinguishing from traditional software systems in terms of structures, functions and scope, designing the Internetware applications should consider these attributes—coordinative, autonomous, evolvable, situational and trustworthy [12]. Meanwhile, as an emerging paradigm, Internetware also faces many challenges, such as

model design, computing environment configuration, developing approach and developing the software system in the paradigm, etc.

This section has discussed the attributes and the challenges of designing the Internetware application. In order to specify how the Internetware works in the Internet computing environment and fulfil the evolvable attribute of Internetware paradigm, Section 3 will propose the prototype of Internet application based Internet computing service, and describe how to integrate software-reengineering techniques into Internetware paradigm..

3 Reengineering Techniques For Internetware Applications

3.1 Software Reengineering

As a combination of reverse engineering and forward engineering, software-reengineering technology [13, 14] is a practical solution for the problem of evolving existing computing systems. Dynamic change management of software systems has been largely performed by using ad hoc techniques that are normally rather expensive and difficult in some cases. There are at least two advantages of using formal methods as the foundation of software reengineering. First, formal design methods can help software developers to acquire enough function description and original design of the system being reengineered, therefore greatly improving the performance of the new system. Second, keeping autonomous and evolvable is one of the key goals of reengineering. By applying and modify the traditional methods, it may be possible to keep autonomous change in the process of reengineering. From the enterprise perspective, reengineering model will be inevitable during the business process. The reduction of cost and time and the improvement of the software system quality can make the enterprise develop faster than other ones.

With the rapid development of economy and science, it is very important to provide the newest and the most suitable service to users. Therefore, this paper considered to keep Internetware evolvable according to the analysis of service-oriented requirement. In this paper, in order to adapt the reengineering process more evolvable, an additional factor—Service Requirement Analysis, has been added in the software reengineering process in Fig. 2, which can make the traditional process more robust.

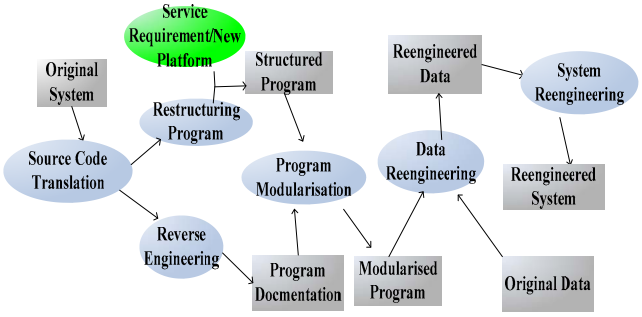

Figure 2. A More Robust Reengineering Process of Software System

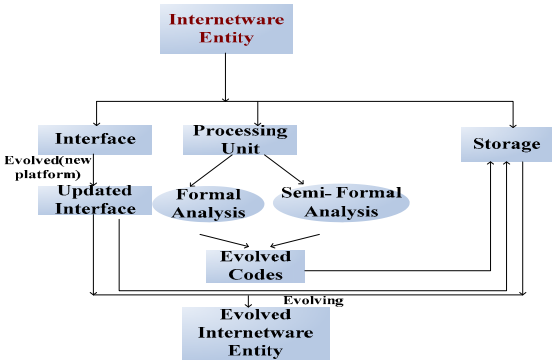

Figure 3. The Process of Application Evolution in Internetware

Fig. 2 illustrates a robust reengineering process. The input to the process is a legacy system or program and the output is a restructured, modularised version of the same system or program. In the process of reengineering, the requirements of computing environment and service, different types of platform and the feedback from enterprises and users have played a very important role in the software development today. For this reason, the developers will improve the original system by putting the function interfaces of emerging computing platform paradigm and service requirements by end users in this restructuring step. Because of these prior operations of the developer, the legacy systems can become more suitable in emerging computing platform after the evolving process. The reengineering system will be polymorphic, autonomous and flexible to apply. In additional, the manual updating is still needed because the open, dynamic and uncertain

computing platform environment. Therefore, combining the autonomous and manual reengineering process will ensure fewer mistakes happened.

3.2 The Proposed Approach

This paper will discuss the process of Internet entity evolution based on service-requirements. As shown in Fig. 3, the chart describes how software systems evolve in Internetware when they receive the service requirement from the enterprise and end users. In this chart, Internetware entity can be recognised as a type of assistant service tools for development and deployment of software application and service. Firstly, end users can send a calling signal to the services and applications through the Internet, the message response will keep loop among users, applications, Internetware entity and Internet. Secondly, if there is nothing to change and require, the requirement will be executed directly. Thirdly, if the end users need some new service or new requirements for the software applications on the Web, due to most of the software service executed with the help of the third entity, the developers just need to do some changes on these entities. Internetware have been thought as the paradigm that collaborated the software service. Therefore, modifying and improving the service in Internetware will resolve the updating issues. When the Internetware receives the updating message called from the software application, it will consider adapt itself according to the new requirements. The Internetware entity can be added a few function interfaces and management beforehand. However, the uncertain and dynamic requirements in the computing environment will also need the manual operations during the updating phase. In this chart, the manual updating has been considered, the Internetware entity will become the reengineered one after the reengineering process. Finally, the task will be executed.

Next, how to deal with the service requirement on Internetware will be introduced. Fig. 4 has shown the evolution schema. In this figure, from requirements to codes, the Internetware will update the Internet entity after the operations.

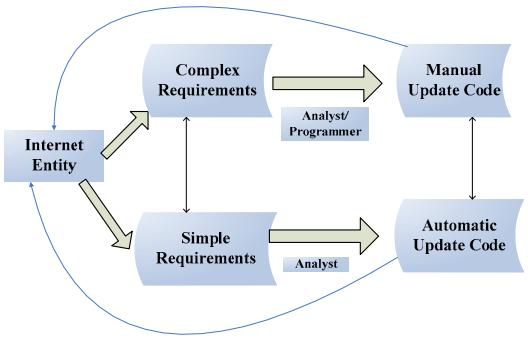

Figure 4. Internet Entity Evolution Schema

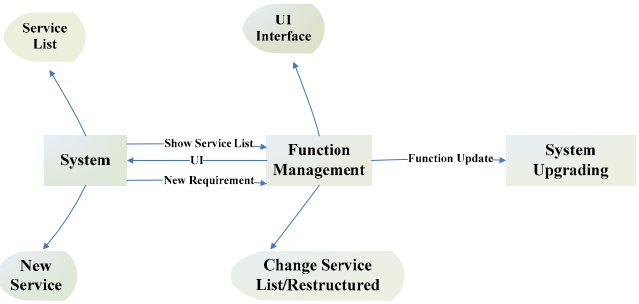

Figure 6. System Upgrading Process

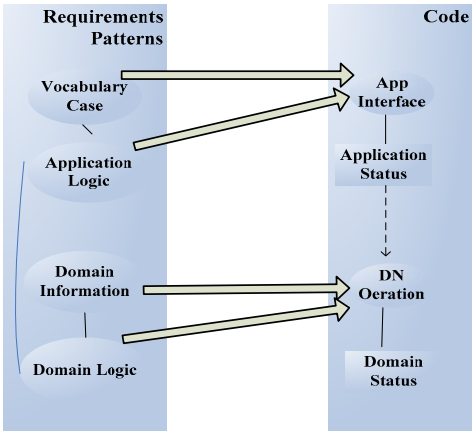

Figure 5. The transformation Rules between Requirements and Code

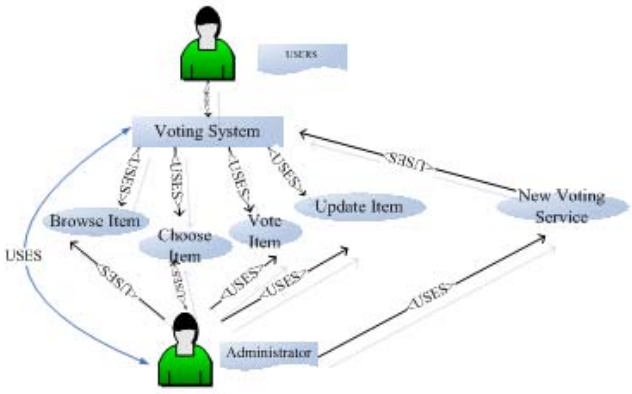

Figure 7. Part of Use Cases for IVMS.

The Fig. 5 describes the transformation rules. After the codes have been updated, the software systems will also generate changes and complete the evolution. Fig. 6 shows the change process of the software systems.

4 Case Study

To illustrate the proposed approach, an example project "Internet Voting Management System (IVMS) ", is presented in this section. This system can be used as different item to vote on the emerging platform.

At the stages of system analysis and system design, use case is still a useful tool to capture user requirements and system modelling. The use cases analysis involves reading and discussing with the potential users of this system. Fig. 7 shows a use cases diagram for IVMS. In order to simplify the process, only parts of use cases are displayed.

At the stage of system evolution, the IVMS will utilise the proposed approach to update itself while the new voting item requirement. The system management will take the automatic and manual updating operations according to new service requirement. Fig. 8 shows the system change based on new service requirement. While users require a new voting item, the message will be packed into grammar glossary and sent to manager, manager checks the service first, if it can be executed directly, the service can update automatically, or the managers will change the codes according to the message come from user interface.

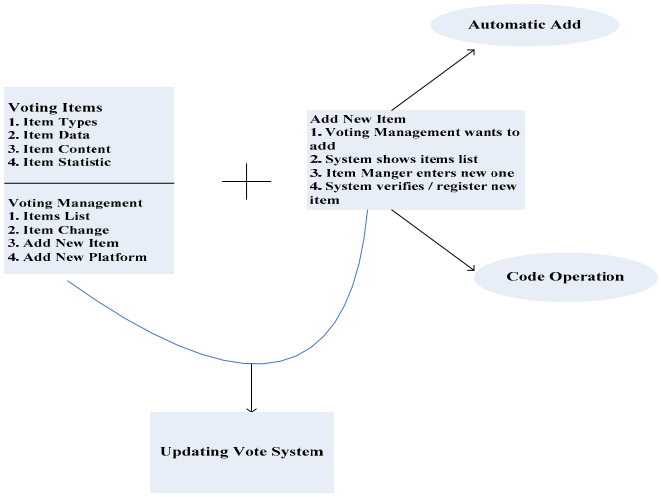

Figure 8. System Change Based on Service Requirement

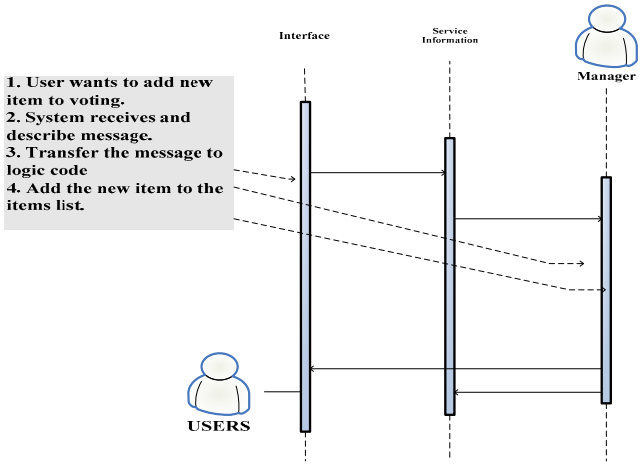

Figure 9. Transfer Process

Fig. 9 shows the process of transferring service requirement into logic code.

The adding items code can be shown in Fig. 10.

```
<body>
<form id="form1" runat="server">
<div class="MainBox">
<span id="menu"><div class="BigMenu">Add Voting Item  : iphone
 <asp:Label ID="Label1" runat="server" Text=""></asp:Label><asp:Label
ID="lbQ_ID"   runat="server" Text=""></asp:Label> </div></span>
<div class="Content">

<table width="100%"  border="0" cellpadding="5" cellspacing="1"
class="MainTable">
 <colgroup span="2">
 <col class="tdbg1" />
<col class="tdbg2" />
 </colgroup>
 <tbody>

  <tr>
 <td>Voting Type: </td>
  <td><asp:TextBox ID="txtI_Title" runat="server" CssClass="longInput
Width="300px"></asp:TextBox>
 <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"
     ControlToValidate="txtI_Title" Display="Dynamic"
ErrorMessage="*">*
 </asp:RequiredFieldValidator>
 </td>
 </tr>
  <tr>
 <td class="style1">Voting Descirption: </td>
<td >
 <asp:TextBox ID="txtI_Memo" runat="server"  CssClass="longInput"
   TextMode="MultiLine" Height="90px"></asp:TextBox></td>
```

Figure 10. Code Description of Adding Voting Item

```
<?xml version="1.0"?>
 <results question="Who is your favorite entry?">
     <votingitem>
         <entryname>iphone</entryname>

   </votingitem>
     <votingitem>
         <entryname>Blackbarry</entryname>
       <votes>16</votes>
   </votingitem>
    <votingitem>
       <entryname>Nokia</entryname>

   </votingitem>
   <votingitem>
       <entryname>HTC</entryname>

   </votingitem>
   <votingitem>
       <entryname>Samsung</entryname>

   </votingitem>
</results>
```

Figure 11. XML description

Meanwhile, in order to keep the updated data, the XML [15] will be utilised to transfer and save even if the IVMS is run in different operating systems and platforms. As descriped in Fig. 11.

With the proposed process and new service requirement, the IVMS can keep evolvable in the Internet computing environment. As an entity in Internetware, its changes will also make the paradigm update.

5 Conclusion

This paper introduces the Internetware paradigm for Internet computing environment, summarises the features and challenges of the Internetware, and then describes the software reengineering techniques and gives more robust reengineering process. The main achievements of the paper proposes how to utilise service requirements analysis approach of software reengineering techniques to make sure that the structure and function of the application in Internetware can meet the dynamic change in the Internet computing environment.

Although a proposed approach to setting software evolution in Internetware, many issues need to be addressed in the future. For instance, make the approach perfect; the research is quite recent and has some distance from resulting into completely automatic process, the model design of Internetware, the operations among the entities in the Internetware and the related experiments on different platforms are still needed.

6 References

[1]. H. Yang, M. Ward, Successful Evolution of Software Systems, Artech House, Boston, USA&London, UK, 2003.

[2]. I. Foster, C. Kesselman, The grid: blueprint for a new computing infrastructure. Morgan Kaufmann, Amsterdam; Boston, 2004.

[3]. M. P. Papazoglou, D. Georgakopoulos, Service Oriented Computing Communications of the ACM, October 2003.

[4]. M. Weiser, The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev., 3(3):3–11, 1999.

[5]. B. Hayes, "Cloud computing, Communications of the ACM, vol. 51, no. 7, pp. 9-11, Jul. 2008.

[6]. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zahariaet, Above the Clouds: A Berkeley View of Cloud Computing, Technical Report, http://radlab.cs.berkeley.edu/, February 10, 2009.

[7]. C. You, M. Zhou and H. Mei, "Towards a Well Structured and Dynamic Application Server," Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 09), Seattle, Washington, USA, Jul. 2009.

[8]. F. Yang, J. Lv, H. Mei, Technical framework for Internetware:An architecture centric approach. Science in China Series F: Info. Sci., 2008, 51(6):610-622.

[9]. J. Lv, X. Ma, X. Tao, F. Xu, H. Hu, Research and progress of Internetware. Science in China Series F: Information Sciences, 2006, 36(10): 610-622.

[10]. O. Raz, M. Shaw, "An Approach to Preserving Sufficient Correctness in Open Resource Coalitions," .In Proceedings of the 10th International Workshop on Software Specification and Design , Shelter Island, San Diego, California, USA, November 5-7, 2000.

[11]. NSF Workshop on "Cyber-Physical Systems", Proceedings available online at http://varma.ece.cmu.edu/CPS, Austin, October 16-17, 2006.

[12]. H. Mei, "Internetware: Challenges and future direction of software paradigm for Internet as a computer," In Proc. the 34thIEEE Annual Computer Software and Applications Conference (COMPSAC 10), Seoul, Korea, Jul. 19-23, 2010, pp.14-16.

[13]. R. Arnold, Software Reengineering, Los Alamitos, CA: IEEE Computer Society Press, 1993.

[14]. E. J. Chikofsky, H. James, Reverse Engineering and Design Recovery: A Taxonomy, IEEE Software 7(1): 13-17, 1990.

[15]. F. Keienburg, A. Rausch, "Using XML/XMI for Tool Supported Evolution of UML Models," In the Proc. of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34), Maui, Hawaii, January 3-6 2001.