

Program Design in Cloud Platform Based on Prototype Development Approach



Tse-Chuan Hsu¹, Dong-Meau Chang², and Yao-Hong Tsai^{3*}

¹ Department of Information Management, Hsuan Chuang University
Centre for Creative Computing, Bath Spa University
Hsinchu 300, Taiwan, ROC
davidhsu@hcu.edu.tw

² Department of Journalism, Hsuan Chuang University
Hsinchu 300, Taiwan, ROC
morgan@hcu.edu.tw

³ Department of Information Management, Hsuan Chuang University
Hsinchu 300, Taiwan, ROC
tyh@hcu.edu.tw

Received 22 June 2015; Revised 11 August 2015; Accepted 22 November 2015

Abstract. For the software system development in cloud, it is difficult to have a complete specification of the desired system. Due to the product development time (time to market) will affect the demand and market trends systems, and in product prices, reducing system development time will enhance product value. Therefore, in this study, we have adopted a prototyping method by cloud-based system development to shorten development time. We plan to found out that such methods can be used to quickly generate the complete system, but the resulted system is still missing some important features or related files. In this paper, a UI-Driven approach was applied on the prototyping method. The methodology is illustrated with the help of a case study on Heroku cloud platform. We discussed and proposed the program design on the applied prototype which deployed edcloud system of Heroku cloud platform and in addition to shorten the development time for developers and customers in communications. The prototype can be developed through a storyboard and UI (user interface)-Driven approach to provide a more accurate and fast system development process. The case study showed that the proposed method was efficient for user to develop a software system in cloud platform.

Keywords: cloud platform, Heroku, Prototyping, Software system development

1 Introduction

Much attention has focused on techniques to bring products to the market more quickly for recent decades. Cross functional teams are important for developers to accomplish successful fast cycle development [1-2, 12]. In recent years, due to the rapid development of cloud software service platform, many software companies have invested in the development of cloud services platform. The demand for software deployment and application software systems is getting bigger and bigger. In the meanwhile, complexity and heterogeneity of software system are also increasing [13]. It's a new problem to complete the system prototype in cloud quickly.

Every cloud-based tool for developers has its own development processes and characteristics [3, 14]. Due to many different ways of the development services, the choice of developer tools is not easy for individuals to learn how to code and to build new cloud or mobile Apps in a short time. However, the

* Corresponding Author

method and processes to produce an application system in cloud are already fixed for every developer tool.

In this paper, we tried to develop a set of software engineering method to integrate developer tools in cloud service from different provider. Based on the proposed, developer can deal with possible effects. For example, additional working time may delay the development schedule, budget overrun may result in poor quality of software and soon. The causes of these problems are that the technology of software development in cloud from the current IT(information technology) industry is not yet mature. As a result, the development of a software system needs to include experts in various fields (domain expert) [8-9] to ensure that the software is in line with the demand. If methods, processes and supporting tool of software development system cannot be integrated with each other, it will not obtain the desired performance.

In this paper, a UI(user interface)-Driven approach was applied on the prototyping method. The methodology is illustrated with the help of a case study on Heroku cloud platform. We discussed and proposed the program design on the applied prototype which deployed cloud system of Heroku cloud platform and in addition to shorten the development time for developers and customers in communications. The prototype can be developed through a storyboard and UI-Driven approach to provide a more accurate and fast system development process.

2 Background and Related Work

System reduce development process largely is an important consideration for the development of cloud software program since we need to curtail the development time, service cost, or other external factors for soft development. In this article, we describes the separation between the user and the developer tools by an interface-driven approach combined with software storyboard mode, interface and programming concepts, and use of the domain connection management technology, management methods, resolving domain name restrictions, and the context of the development. For developers, the UI provided faster solutions to deal with the above criterion.

2.1 UI design

In order to meet different requirements, software designer needs to develop of different kinds of systems and to follow the most important specifications of users. Therefore, the UI is the first consideration of a prototype development method when different users expected different interface, specially unique interface for their owned. In the initial phase of system development, the developer construct a customized system according to the different needs from different users, i.e., user requirement [10-11]. In the system requirement phase, we needs to design the interface and image icon for developers and users. The developer can then extends the interface to class, methods and the class diagram [4-7]. For the development of a cloud application service, one can design the process and dynamic model of the system that fitted the user's demand by UI-driven user interface design method in the requirement phase. Furthermore, the developer extended categories, methods and related modeling class diagram of the model through the process and the UI-Driven by design. That is more clear and efficient than the traditional development methods in the web-based service platform.

2.2 Storyboard

When making a movie, an animation, a television show, a music video or a commercial, a series of illustrations that briefly summarize the key scenes and all events of them is called a storyboard [8]. That was usually done before the film is actually shot. Illustrations describe the essential details needed to communicate the scene-specific information and include the use of graphical notations and additional text descriptions. Storyboard is a fast and economic way to express the idea, mirror mode, the length of time, dialogue or special effect of what the visual media is going to look like before actually shooting. While in the present study, we applied the concept of storyboard technology. The technology used in the movie storyboard to describe the story is now using pictures to tell the user's needs, like "When I saw it I knew what I want" (IKIWISI) [15]. In general, this phenomenon exists in all software development. Therefore, the UI are designed by this relationship between user and software system. Visual communication are used to confirm the needs of the user and let user know the system process.

At present product markets, tool kits for software development needs to have the relevant software tools, such as SoftScore. It identify and establish tools to assist the user to confirm the process and needs. It usually provide simulate functions which are similar to the way of playing a movie. It also allows users and developers to confirm the appearance and functionality of the system. That is easier to understand than the traditional file-based expression mode. SoftScore comply with TDD (test-driven development), can be annotated test method and system functions, export demand model becomes PDF file, output Use Case of UML (unified modeling language) and test cases.

2.3 Prototyping

For the software system development in cloud, it is difficult to have a complete specification of the desired system, especially for that required the time to market issue. In the cloud service platform, we usually obtain a complete and accurate specification of the system by the template design, providing users a quick reference system screen, and gradually corrected process. It is useful especially for some of the time to market software system development. Since users often have only a vague idea of the system, it is difficult to express completely and accurately overall requirement on the system. Developers is also unclear to the problem to be solved or the user's demand.

With the increasing of development functions, users may have new demands. Since the new system environment has changed, the system is also asked to change as the user requirements. Developers may encounter some problem that are not expected in the original design stage of the process. It is difficult to pre-describe the specification about the changes of user's demands in advance. Similarly misunderstanding in communication will also become the developer's obstacle to the success of the system.

In this study, we focused on software engineering and discussed about prototype development life cycle. Prototype development life cycle was divided into five stages. First, perform the initial analysis. Secondly, define the prototype objective. Thirdly, specify the prototype. Fourth, construct the prototype. Fifth, evaluate the prototype and recommend change (see Fig. 1).

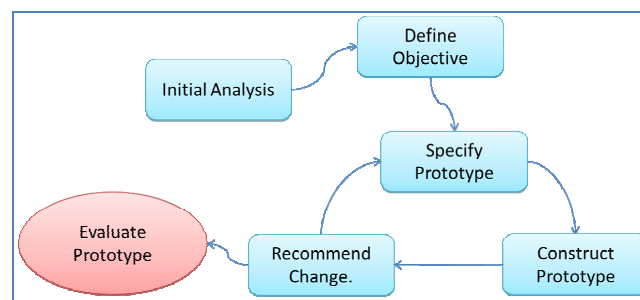


Fig. 1. Prototype development life cycle

3 The System Architecture

In this paper, we presented the system architecture to explain the relationship between the development environments. Each subsystem is shown in Fig. 2. In the stage for finding user's requirements, developers may have the customer's needs like views, events, and actions by UI description through the way of the script. After making the decision in the script, the user can set the system elements, the attribute element, and the element type by the appropriate related components in accordance with the instructions to complete the system. Then components and event information will pass to the responsible subsystem for corresponding actions. In this stage, information will first send to the database, if objects are appropriate, they may be retained in the database. The responsible subsystem of connection for action will display the information of components and events through the approach of the separation between before the scene and behind the scene. For the details of the component settings and the connection relationship, they will be done behind the scenes by setting parameters. The simulated virtual environment will then be obtained by compiler of the development environments. Finally, confirmation have to check the user's needs and the script. If it is correct, that means the resulting system is correct.

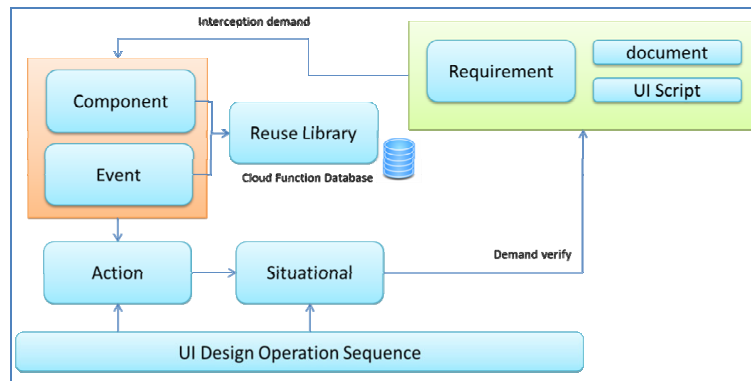


Fig. 2. System Architecture

4 Research Methods and Prototype Development Process Storyboard

For the prototype development in cloud base systems, we applied the Python language and Heroku platform combined with cloud application architecture for our research. First, situational settings of the storyboard can determine the relevant part of the required elements and components for system development. By the reuse methodology, one can easily load the required components from the previous system and construct a system prototype. Situational simulations can help users confirm the system prototype. The storyboard prototype developing process is shown in Fig. 3.

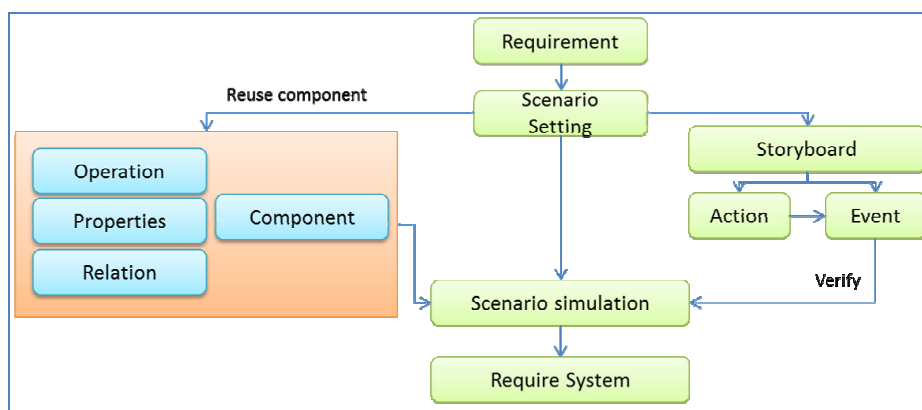


Fig. 3. Prototype development process storyboard

Before the system development, developers and users have to complete the identification of system needs. That is extremely important and will result in reducing post-modification of the system development and the development gap between developers and users. Through the pattern of the system prototype, the display will demonstrate and discuss with the user dynamically, such that system requirements can be more clearly communicated and confirmed with the user. Therefore, we try to carry out the part of system requirement which input by the way of visualization tools. The input process was done by just clicking through the interface between the user and the system. Then, the system will automatically analyze the components, actions and events from the user’s input to complete the development process.

4.1 Reuse component

There are many approaches for component reuse in different applications, e.g., copying and pasting, inherited class, framework architecture, application programming interface, etc. In this study, we used the combination of cloud services platform to implement the prototype development approach and each system function was treated as a component. Each component has its own attributes, operations, relationships and reuse class. Component functions includes sending e-mail, storing messages, attach files and others.

Operations are the functions of system elements and have the corresponding basic specifications but

each of them does not have the real entity. Some of the properties of a component is the name, color, size and other characteristics. Hierarchical relationship with other elements contains separated class relations and class inheritance relationship.

4.2 Situational Setting

About the arrangement of situational setting, the relationship between the object and the event will be represented by the way of the finite state machine. The object selected by the user will transfer from the current state to the next state. There is a dynamic link between two states, which is the produced event. Each object will inherit the characteristics of the previous component by the concept of component reuse and adding or deleting attributes are also available to the new component. The relationship of component reuse is shown in Fig. 4.

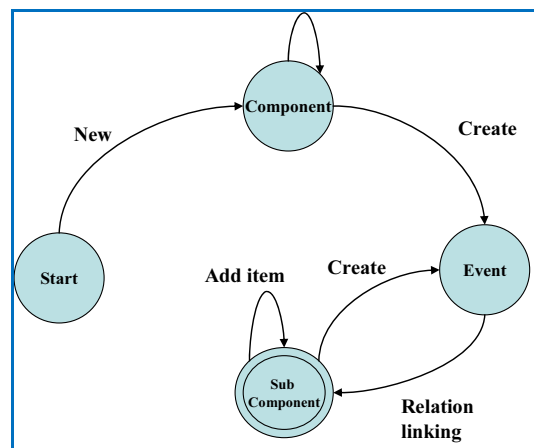


Fig.4. Component reuse

4.3 Scenario storyboard in cloud development

In the past, we had to understand the usage of different API (application programming interface) and apply the system development process and project correspondence when engineer needed to establish a set of software applications since they must be converted to the corresponding event in system functions. Concerning about the software development, the delay time will result in higher investment costs since new developers need to understand many learning tools before he is going to modify the system.

In this study, we developed an email system in cloud like Heroku system by the storyboard concept. We tried to work with the requirement for developing capabilities include a script for a story. The story of the script of the system operation functions was scheduled and established research on component testing part. The system development through a script is like to select in Lego model. Each component automatically cascaded its corresponding API. It adapted NoSQL (not only structured query language) data form in the database setting. Developers do not need to gradually develop by the basic system. They directly construct the system through the cloud service PaaS (platform as a service) platform, combined with the script storyboard. Each individual element can be built by component reuse.

We show our studies on e-mail service developing items in Table 1. In order to send and receive e-mail functions. It commonly illustrates how to use these principles to quickly complete the development functions. The first one is the sending e-mail functionality, although there are many available functions in the development process in accordance with the instructions in Heroku platform API add-on. Optional components can be provided by different developers, like receiving e-mail function. For receiving e-mail, one must include more steps of the basis of cloud based system to complete the function. Heroku did not include a system function for user to build an e-mail service solution. For the DNS MX records, development must rely on other mail service providers to let the letter send to Heroku platform. Furthermore, we used the platform for handling incoming mail, and defined all function to story items. When users can select different developing function according to their own needs, the creation of the web e-mail service systems will be done. Since they choose component on the storyboard about e-mail items in the system, the system will automatically create the corresponding web code, database system, and functions embed-

ded in the system that they tried to develop. Table 1 shows the corresponding component architecture of mail service developing items.

Table 1. Mail service developing items

Story board items	Reuse component *	Cloud based API
Sending email	3 rd written libraries for sending mail under Heroku platform	Heroku add-ons
Email receiving	Mail bonding provided by mail service vendor Binding to Heroku platform by 3 rd written libraries Display status	ISP such as Google mail Heroku add-ons Heroku components

Note. *: defend the component on the cloud systems helping with recuse in different story board development.

5 Case Study

In the proposed result, we applied the cloud platform from Heroku cloud system to perform our case study. It is a new platform from salesforce solution. In cloud platforms, users need to code a program to integrate different component work together, like the System Integration Company, but from different requirements, if we can build up all of system’s component, we can ease to reuse different components to build up the prototype system in software system development market. In the meanwhile, developed the component of the reuse architecture for software engineering life cycle is much important in cloud systems.

We built the email service for the case study in this research. Traditionally, we knew that SMTP (Simple Mail Transfer Protocol) and POP3 server can provide the email services. However, one cannot implement the service by installing the whole environment of email server in the cloud platform. In this case, we showed how to design a prototype development environment base on storyboard process to provide email service.

We set up the composition of design elements which can be discovered in a cloud environment and used repeatedly. After the user surveyed and confirmed all elements in the website, it must be ready for the email sending and receiving functions. So, in development workflow, we needed to build up different components, like sending mail, receiving mail and bonding for mail address. About sending e-mail function, we can use add-ons items from Heroku platform since there are some components can be reuse (see Fig. 5).

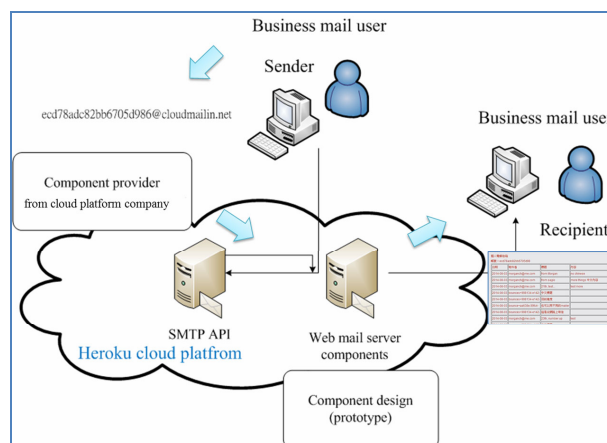


Fig. 5. Case study architecture

In general, the function can be used to send a letter by the following two ways: (1) SMTP, mode (2) API mode. If you are using the former mode, it is borrowed from Heroku platform as intermediary (middleware). As long as the specified SMTP server connection status and related information are given, you can use different provider’s service to send an e-mail message without any coding process. From system program developer’s point of view, deals with sending e-mail process was the main goal but not coding. We developed the components for a general purpose on mailing treatment. The following program frag-

ment was based on Python/Django architecture to send a letter written in the Heroku platform.

```

from email.mime.text import MIMEText
sent = request.POST['sentemail']
subj = request.POST['title'].encode('utf-8')
sentHtml = request.POST['mailcnt'].encode('utf-8')
import smtplib
smtp = smtplib.SMTP("smtp.mailgun.org", 587)
smtp.login('xxx.mailgun.org', 'password_from_mailgun')
# typical values for text_subtype are plain, html, xml
text_subtype = 'html'
msg = MIMEText(sentHtml, text_subtype)
msg['Subject']=subj
msg['From']=abc@abc.com'
msg['To']=sent
msg = smtp.sendmail(abc@abc.com', sent, msg.as_string())
smtp.quit ( )

```

The following settings are the SMTP server parameters.

```

#using cloud mail server : sendgrid
EMAIL_HOST = 'smtp.sendgrid.net'
EMAIL_HOST_USER = 'appxxxx@heroku.com'
EMAIL_HOST_PASSWORD = 'yourpasswd'
EMAIL_PORT = 587
EMAIL_USE_TLS = True

```

In the API mode. We use the API approach in general, you can communicate directly with the host, so the effectiveness of the way through the SMTP server than a lot better. To find relevant information and API example of how to write a program, in this case, we try to design using API design the component can be reuse in other prototype system. Also, before write processing program, to the correlation function hall (e.g. API Library) fitted to their environment. Various programs language and development framework can be used, there are different installation methods, refer to the documentation and installation instructions on the official website. We use Python / Django framework, so need to match the corresponding system API library installed.

The following example is the use of mandrill API Library, for example, the user filled contents of the letter sent directly through the API.

```

import mandrill
mandrill_client = mandrill.Mandrill('API key')
message={}
message['from_name']='morgan chang'
to=[]
toDir = {}
toDir['email']=request.POST['sentemail']
toDir['name']='Recipient Name'
to.append(toDir)
message['to']=to
subj = request.POST['title'].encode('utf-8')
message['subject'] = subj
sentHtml = request.POST['mailcnt'].encode('utf-8')
message['html'] = sentHtml
message['from_email']='receiver mail address'

```

```
result = mandrill_client.messages.send(message=message)
```

The testing messages collected on the web server is shown in Fig. 6 and Fig. 7



Fig. 6. Dashboard transition to UI information system from user define

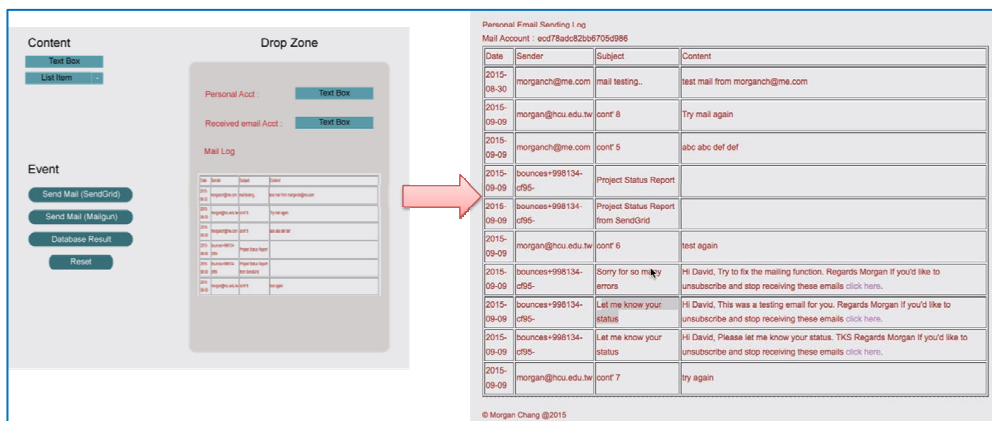


Fig. 7. The prototype from cloud platform mailbox solution

6 Conclusion

Due to the market requirement will largely affected the system development time, reduce development time in system will also reduce the overall development cost and improve product value. Therefore, in order to shorten the development process, we use the prototype development methods combined with cloud applications form different companies providing service application component to integrate the system development. A UI-Driven approach was applied on the prototyping method for software system development. The proposed methodology is illustrated with the help of a case study on the Heroku cloud platform in this article. We proposed the program design on the applied prototyping development of the case study to show how to deploy cloud system of Heroku cloud platform and in addition to shorten the development time for developers and customers in communications. The prototype can be developed through a storyboard and UI-Driven approach to provide a more accurate and fast system development process.

Acknowledgement

This paper was supported by a grant from National Science Council MOST 104-2221-E-364-002 and Hsuan Chuang University.

References

[1] M. Almosry, J. Grundy, R. Sadus, W. van Straten, D.G. Barnes, O. Kaluza, A suite of domain-specific visual languages for scientific software application modelling, in: Proc. Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium, 2013.

- [2] E. Guzman, Visualizing emotions in software development projects software visualization (VISSOFT), in: Proc. 2013 First IEEE Working Conference, 2013.
- [3] T. Mastelic, I. Brandic, A. Garcia, Towards uniform management of cloud services by applying model-driven development, in: Proc. Computer Software and Applications Conference (COMPSAC), 2014.
- [4] M. Szlenk, Formal semantics and reasoning about UML class diagram, in: Proc. International Conference on Dependability of Computer Systems, 2006.
- [5] T. Clark, A. Evans, P. Sammut, J. Willans, Applied Metamodelling: A Foundation for Language Driven Development, Xactium, London, 2004.
- [6] C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, Prentice Hall, Englewood Cliffs, NJ, 2001.
- [7] A.B.H. Ali., F. Boufares, A. Abdellatif, Checking constraints consistency in UML class diagrams, Information and Communication Technologies 2(2006) 3599-3604.
- [8] J.S. Jwo, Y. C. Cheng, Pseudo software: a new concept for iterative requirement development and validation, in: Proc. Software Engineering Conference, 2007.
- [9] M. Lindvall, P. Donzelli, S. Asgari, V. Basili, Towards reusable measurement patterns, in: Proc. Software Metrics, 2005.
- [10] N. Maiden, User requirements and system requirements, Software 25(2)(2008) 90-91.
- [11] L.W. Bezanson, L.G. Fields, D.O. Knight, M.J. Merritt, B.R. Millard, D.S. Miller., P.R. Rony, Engineering support system user requirements, Micro 5(5)(1985) 36-51.
- [12] B.J. Zirger, J.L. Hartley, The effect of acceleration techniques on product development time, IEEE Transactions on Engineering Management 43(2)(1996) 143-152.
- [13] S. Frey, F. Fittkau, W. Hasselbring, Search-based genetic optimization for deployment and reconfiguration of software in the cloud. In: Proc. of the 35th International Conference on Software Engineering (ICSE 2013), 2013.
- [14] J. O'Dell, Top 10 cloud-based tools for developers. < <http://venturebeat.com/2013/08/01/the-top-10-cloud-based-tools-for-developers/> > , 2013.
- [15] B. Boehm, Requirements that handle IKIWISI, COTS, and rapid change, Computer 33(7)(2000) 99-102.

