

A Technique of Business Model Driven Analysis and Test Design for Developing Web Applications

Yongsun Cho¹, Woojin Lee², Kiwon Chong³

¹Department of Computing, Soongsil University, Seoul, Korea
yongsuns@hanafos.com

²Department of Computing, Soongsil University, Seoul, Korea
bluewjl@dreamwiz.com

³School of Computing, Soongsil University, Seoul, Korea
chong@comp.ssu.ac.kr

Abstract. The technique of creating analysis models of a web application from a business model is proposed for the rapid and efficient development. Moreover the technique of creating test cases from a sequence diagram for a web application is proposed. The web page list are created from business model which is depicted using the notation of UML (Unified Modeling Language) activity diagram. The page diagram and logical / physical database models are created based on the web page list and extracted data fields. Test cases for web application are extracted from call messages (including self-call messages) of UML sequence diagram. The efficiency of these techniques has been shown using a practical case study which is development of web application for RMS(Research material Management System).

1 Introduction

Recent business environments have been changing into Internet business environments and web applications have been developed continuously for advertisement, sale of goods, customer support, etc. in Internet business environments [1]. Web applications become more complicated, because more various requirements should be accommodated by web applications according to achieving various business using web applications. To succeed in these environments, systematic and effective analysis should be done ahead to construct. However, in most cases of web application development, there are difficulties in looking for a suitable analysis method, so developers jump directly into the implementation or they have difficulties with development methods of a large scale program like RUP [2]. Some studies[3,4,5,6] are performed but they concentrate certain portion and don't support whole analysis tasks. Therefore a suitable method and techniques for development of web applications is still required.

Moreover the accuracy of web applications is also emphasized in these web environments. If the web application doesn't operate correctly or it discontinue because of wrong operation, corporations must suffer a great loss and be notorious. To prevent

* This work was supported by the Soongsil University Research Fund.

beforehand these situations, it is necessary to test the reliability of web applications. Although many techniques for testing web applications have been being studied, those are not enough. Most of early techniques for testing web applications check syntax and context of html, jsp, asp files or check correctness of links among them. Furthermore most of recent techniques for testing web applications check the operations of single web page or check the call-relations among web pages. The clustering test for testing collaboration of web pages is required for reliability of web applications as well as these.

Therefore, this paper proposes business model driven analysis and test design techniques for rapid and efficient development of web applications.

2 Business Model Driven Analysis

2.1 The Process of Business Model Driven Analysis

The process of business model driven analysis and test design for development of web application is as followings.

1. Business Modeling
2. Page List Creation
3. Data Field Extraction
4. Page Diagram Creation
5. Database Modeling

2.2 Business Modeling

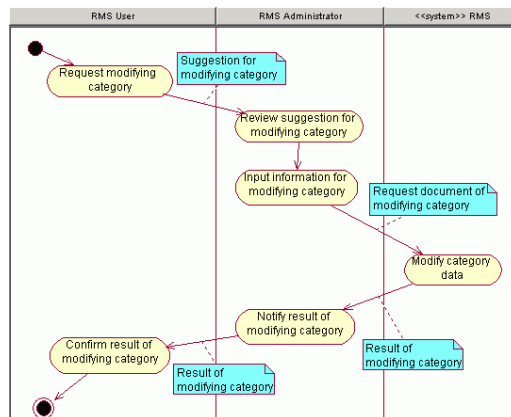


Fig. 1. Business model – modifying category in RMS

The business modeling is used to define the flow of a business using web application. Usually, domain experts develop business models. The business model is depicted using the notation UML activity diagram [7]. Note that, a business activity of

business model is described by verb form. If a role presented by swimlane is a web application, a stereotype called <<system>> is placed in front of the name of swimlane.

First the outlined business model is developed. The outlined business model specifies the whole and rough flow of a business on one page omitting details, and in the next step in the same task, each flow is refined with the business document in detail.

A detailed business model is developed after the outline of the business model. In the detailed business model, each business flows in the outlines of the business model and there it is described in more detailed and business documents for business activity are attached on arrows connecting two business activities. Additional business activities are added if needed.

2.3 Page List Creation

The page list is list of web pages to be constructed for web application development. The page list is created from business documents in the business model. The business documents on the relationship with the <<system>> swimlane in the business model are implemented in the web application, so these documents will be transformed into web pages.

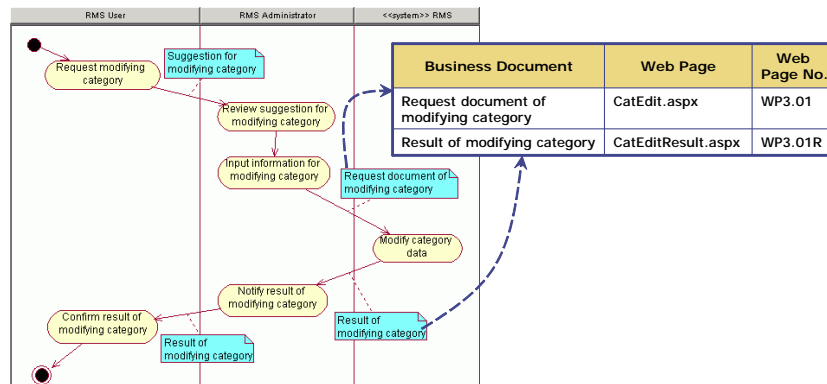


Fig. 2. Page list creation

The page lists are extracted using following technique.

1. Business documents entering to or out-going from the <<system>> swimlane are considered as candidates of web pages. Business documents which are not related with <<system>> swimlane are not performed within the system, so those are analyzed just for understanding business domain.
2. Extracted web pages are analyzed based on real business documents. If there are no documents to be used, the pages are analyzed originally by developer.
3. The extracted pages are refined. One page in the page list can be separated into several pages in actual implementation or some initial web pages can be merged into one page.

4. The additional pages are created. You should consider additional pages for error messages or decorating.

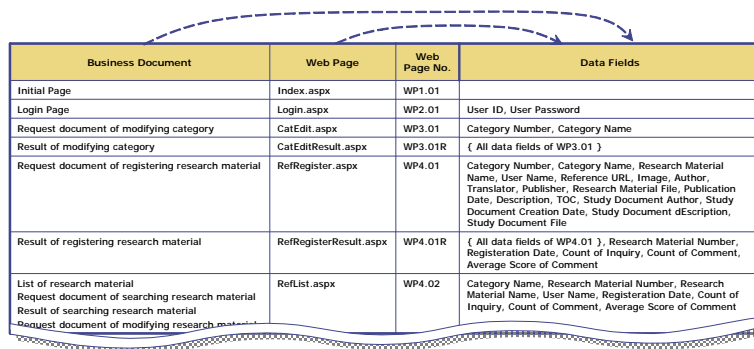
Web pages are extracted from business documents which have relationship with <<system>> swimlane. Some business documents are merged to one web page, and business document are separated to some web pages. It is shown in Fig. 3.

2.4 Data Field Extraction

Data Items in the business documents related with <<system>> swimlane are extracted as data fields for the web page transformed from that document. The other data fields are also extracted with consideration of the web pages extracted in the previous step. The data fields become the elements of user interface for input or output in web browser and realized into elements of FORM tag in the HTML language. After this, the additional data fields are added according to the need.

The data fields are extracted using following technique.

1. Business documents entering to or out-going from the <<system>> swimlanes in business model are analyzed.
2. Data fields are extracted from items on business documents related <<system>> swimlanes. Necessary data is already structured on each off-line business document.
3. Data fields are also extracted from web pages extracted previous task. Many data is located redundantly on business documents and extracted web pages. In this step, data extracted from items only on web pages but business documents.
4. The type and size of data fields are considered according to expression way and processing way.
5. Extracted data fields are refined. Some are separated and some are merged.
6. Additional data fields are considered for accomplishing application logic.



Business Document	Web Page	Web Page No.	Data Fields
Initial Page	Index.aspx	WP1.01	
Login Page	Login.aspx	WP2.01	User ID, User Password
Request document of modifying category	CatEdit.aspx	WP3.01	Category Number, Category Name
Result of modifying category	CatEditResult.aspx	WP3.01R	{ All data fields of WP3.01 }
Request document of registering research material	RefRegister.aspx	WP4.01	Category Number, Category Name, Research Material Name, User Name, Reference URL, Image, Author, Translator, Publisher, Research Material File, Publication Date, Description, TOC, Study Document Author, Study Document Creation Date, Study Document dEscription, Study Document File
Result of registering research material	RefRegisterResult.aspx	WP4.01R	{ All data fields of WP4.01 }, Research Material Number, Registration Date, Count of Inquiry, Count of Comment, Average Score of Comment
List of research material Request document of searching research material Result of searching research material Request document of modifying research material	RefList.aspx	WP4.02	Category Name, Research Material Number, Research Material Name, User Name, Registration Date, Count of Inquiry, Count of Comment, Average Score of Comment

Fig. 3. Data field extraction

2.5 Page Diagram Creation

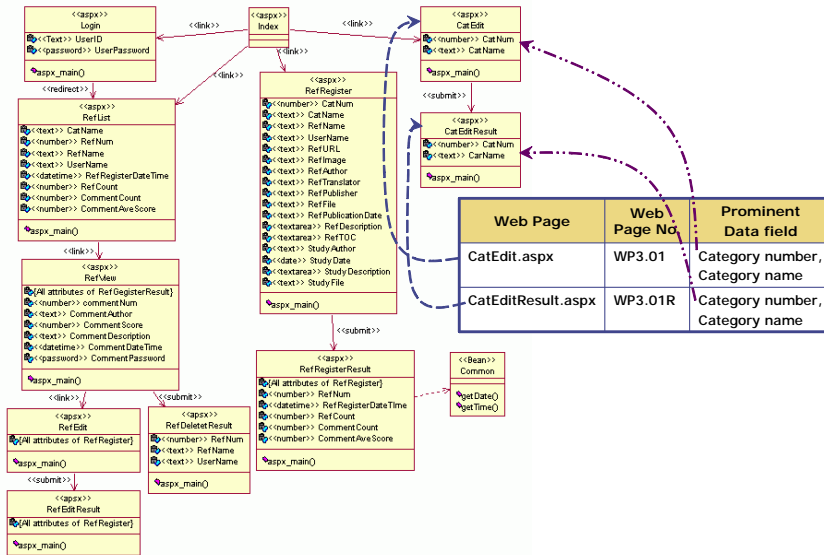


Fig. 4. Creating of a page diagram

Extracted web pages, page logic and data fields of the web pages are presented in the page diagram. The notation of UML class diagram is used for representing the page diagram [9, 10]. The extracted pages are displayed with the notation of class and the relations between web pages are displayed relations between the classes of the class diagram. The stereotypes are used in order to add meaning to the web pages and relationships.

A page diagram is created using following technique.

1. The web pages on the page list are represented in classes. Web pages are distinguished by stereotype such as <<html>>, <<jsp>>, <<asp>>, <<aspx>> and <<php>> according to the way of implementation. You can use another stereotypes according to the standard of your project.
2. The relation between web pages is represented in a relation between classes. The kinds of relation are distinguished by stereotype such as <<build>>, <<link>>, <<redirect>> and <<submit>>. You can also use another stereotypes according to the standard of your project.
3. The data fields to be used in the logic of web pages are represented in the attributes of class. The data fields of web pages are expressed using stereotype such as <<text>>, <<password>>, <<radio>>, <<checkbox>>, <<file>>, <<select>> and <<textarea>> based on the name of element HTML form tag or <<number>>, <<text>>, <<datetime>> based on the type of attribute. Variables to be used in application logic are expressed using stereotype such as <<Bean>>, <<Link>>, <<Select>> and <<Submit>> according to the characteristic of variables.

4. The logic of server pages or script functions is represented in the function of a class. The basis logic of .aspx server page is marked with “aspx_main()”, and script functions are named using each function name.
5. The page diagram is refined.

2.6 Database Modeling

The database model is designed from the extracted data fields. According to the kinds of database such as relational data base, object-oriented data base and object relational data base, the database model can have another form.

The design technique for the database model is described based on relational database, which is mostly used at present. The extracted data fields are grouped into entities in consideration of a relevant web page and logic of the page, and the relationships between entities are defined. The extracted entities and the relationships are implemented by the tables and the foreign keys of the database.

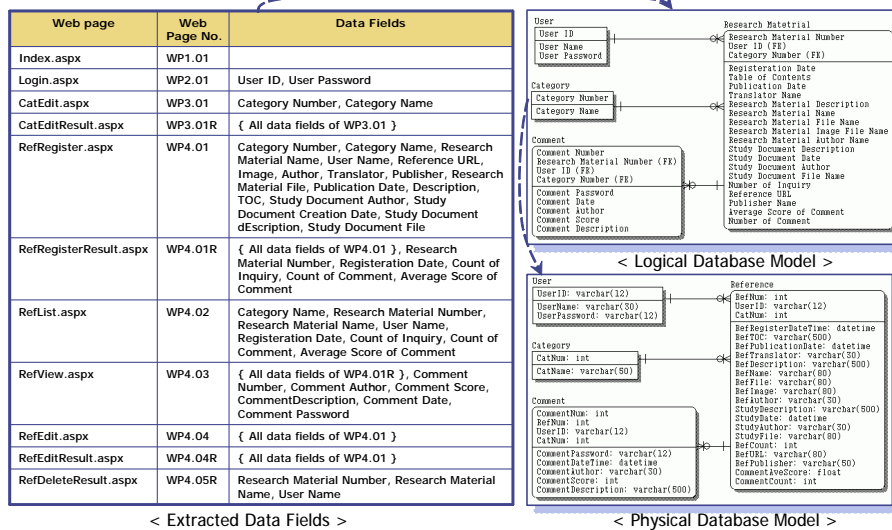


Fig. 5. Database model design

A logical database model is designed using following technique.

1. The extracted data fields are grouped into entities in the consideration of a relevant web page and logic of the page. Each field becomes an attribute of belonged entity.
2. The additional data fields are added to each entity according to the need.
3. The initial ERD (Entity Relationship Diagram) is created by adding the relationships between entities.
4. The ERD is refined with normalization and de-normalization according to the need.

A physical database model is created using following technique.

1. The physical database model is created by duplicating the logical DB model and changing the name of attributes. Each name of attributes of entities is changed into the name to be used in the database.
2. The type and the size of each attribute are determined.
3. The data definition language of ERD for specific DBMS is generated from physical database model with the E-R modeling tool and the database is constructed. Each entity becomes a table and each attribute becomes a column of a table. The relation between entities is implemented as foreign key in database.

3. Test Case Design for Web Applications

3.1 Test Case Design

The test of web applications is achieved from smallest unit test to the whole system test. If a submodule of the module operates incorrectly, the test result of a module is not reliable. Therefore it is necessary to test an application in level-by-level.

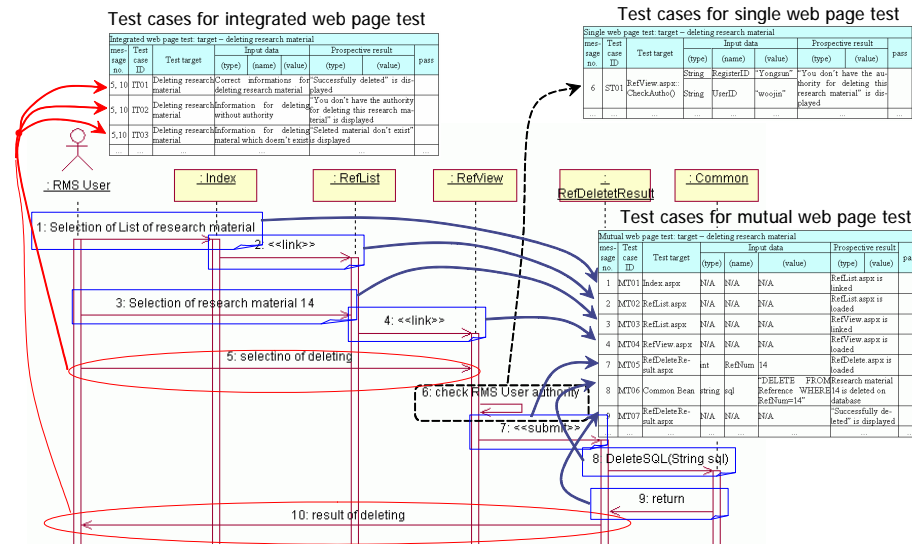


Fig. 6. Test case extraction from sequence diagram

The test of web applications is composed of single web page test, mutual web page test and integrated web page test. Each level of test is performed iteratively because detected failures from the test should be corrected and regression test must be performed in the same environment again for confirming the correct result. Besides this, relative parts of the web application should be tested because changes after testing can affect other parts of the application.

The transfer of messages among web pages can be expressed using the notation UML sequence diagram. The test cases are extracted from sequence diagram of web pages. The test cases for single web page test are extracted from self-call message and test cases for mutual web page test are extracted from the messages between web pages. The test cases for integrated web page test are extracted from the message which are sent by actor to system and received back by actor. The technique for extracting test cases is shown in Fig. 6.

3.2 Extracting Test Cases for Single Web Page Test

First of all, single web page test is performed for testing of web applications. Each web page is tested in the single web page test. Context and resources are examined for static pages like html pages. Context examination confirms whether the page made out according to syntax and resources examination confirms existence of the linked or called URL. These are easily tested with html editors or html syntax checking tools.

The test cases for single web page test are extracted from self-call messages of each page in the sequence diagram. This case is that the server page calls its own script functions or the page is reexecuted by itself.

The test data are added to the test case if script functions or web pages needs input values. Moreover, the values of objects in the form tag of the web page are considered as test data if the script functions reference the values of objects in the form tag using document object [11].

The related web pages, script functions and variables with these test cases are referenced from the page diagram such as Fig. 4.

Table 1 is an example of extracting test cases from sequence diagram of Fig. 6 for single web page test. ST01 is a test case for testing the script function that confirms the authority of user for deleting a research material.

Table 1. An example of extracting test cases for single web page test

Single web page test: scope – deleting research material								
message no.	Test case ID	Test target	Input data			Prospective result		pass
			(type)	(name)	(value)	(type)	(value)	
6	ST01	RefView.aspx:: CheckAuthor()	String	RegisterID	“yongsun”	“You don’t have the authority for deleting this research material” is displayed		
			String	UserID	“woojin”			
...	

3.3 Extracting Test Cases for Mutual Web Page Test

The mutual web page test which examines if the pages are correctly performed in their mutual relation is performed after single web page test. The purpose of the mutual web page test is to examine if a page is linked to another page without loss of information, incorrect transfer of information or error.

The message 5 is a test case and the message 10 is a result of the test case in Fig. 6. Table 3 is an example of extracting test cases for integrated web page test from sequence diagram in Fig. 6. IT03 is a test case to examine if the system announces error to user when a user is going to delete a nonexistent material.

4 Conclusion

A business model driven analysis and test design techniques for developing web applications is proposed and the efficiency of these techniques has been shown using a practical case study which is development of web application for Research material Management System.

It is expected that analysis and test of web application will be accomplished quickly, easily, and efficiently with these proposed techniques. It is also expected that organizations developing web applications should achieve higher efficient skills when developing web applications using the proposed techniques.

The proposed techniques will be refined and will evolve into whole process for developing web application with the help of new case studies and the techniques for design and implementation. Furthermore the test tool for web applications will be developed to support the proposed test case design technique.

References

- [1] Abhijit Chaudhury, et al., Web channels in E-Commerce, Communications of the ACM, Jan. 2001
- [2] Philippe Kruchten, "The Rational Unified process", Addison Wesley Longman, Inc. , 1999
- [3] Kenji Takahashi, Eugene Liang, "Analysis and design of Web-based information systems", Computer Networks and ISDN Systems, vol. 29, pp. 1167-1180, 1997.
- [4] Silvana Castano, Luigi Palopoli, and Riccardo Torlone, "A General Methodological Framework for the Development of Web-Based Information Systems", LNCS 1921, pp. 128-139, 2000.
- [5] Stefano Ceri, Piero Fraternali, Aldo Bongio, "Web Modeling Language (WebML): a modeling language for designing Web sites", Computer Networks and ISDN Systems, vol. 33, pp. 137-157, 2000.
- [6] Jyhjong Lin, Yalei Chu, "Object-Oriented Analysis and Design of Web-Based Information Systems", Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '01), pp. 68-75, Apr. 2001.
- [7] OMG Unified Modeling Language Specification Version 1.4, September 2001.
- [8] John Chssman, et al., "UML Components", Addison-Wesley, 2001.
- [9] Grady Booch, et al., "The unified Modeling Language User Guide", Addison Wesley Longman, Inc. , 1999.
- [10] Jim Conallen, "Building Web Application with UML", Addison Wesley Longman, Inc. , 2000.
- [11] Mashito Hamba, Ryuichi Okakura, "HTML&Java Script Dictionary", youngjin.com, 2000.
- [12] Roger S. Pressman, Software Engineering: A Practitioner's Approach (5th Edition), McGraw-Hill, 2000.