Testing UML Design Model for Web-based System

Mazni Omar[a] & Norliza Katuk[b]
Department of Computer Science
Faculty of Information Technology
Universiti Utara Malaysia
06010 Sintok, Kedah
[a] mazni@uum.edu.my & [b] k.norliza@uum.edu.my

Abstract

UML has been widely used as a modeling tool in software development. Software developed with UML has to be tested to assure its quality and to prevent faults. Current practice in UML design evaluation consists of walkthrough and inspections. However, these techniques are too complex and tedious because the reviewer needs to track large amount of information. Furthermore, the lack of assessment of design quality, and deficiency in detecting and correcting design fault in the model can increase the total software development costs and time to market. This indicates that the need of testing and validating design model for web-based applications is high. Therefore, this study intends to justify whether the DCD criteria can be used to derive suitable test cases for web-based application UML design model.

Introduction

Testing of web applications is a specialized area of software testing and it is quite a new area. Due to the unique characteristic of web applications, conventional software testing tools are not adequate in dealing with web applications. It is critical to develop effective methodologies and tools for testing web applications (Jia and Liu, 2002). Adequate testing is essential to guarantee the quality of a software system and to ensure that the software satisfy the user requirement. Driven by the extreme demands of business world and enthusiasm of the public, more and more businesses being conducted through the web. Thus, testing web application becomes more challenging than conventional software (Jia & Liu, 2002). Therefore to ensure the web application works correctly, the web functionality must be thoroughly tested using an appropriate testing technique.

Object-oriented Analysis and Design (OOAD) technique with Unified Modeling Language (UML) has been adopted to model the requirements for this study. UML which is created by Booch et al. (1998) is a language for specifying, visualizing, constructing and documenting the deliverables of software product. It enables stakeholders that are involved in software development to document and to describe the software in a standard way. This paper will discuss about the process of testing UML design model for web-based application using DCD criteria.

UML Modeling and Software Testing

The Unified Modeling Language (UML) is a visual modeling language that can be used to specify, visualize, construct and document the artifacts of software system (Booch et al., 1998). According to Eriksson & Penker (1999), UML can be applied in different phases of system development, from the requirement specification to the test of finished system. There are many types of diagrams in UML and each diagram in UML has its own specific purpose. Sparxsystems defined a few types of UML models such as Use Case Model, Interaction or Communication Model, State or Dynamic Model, Logical or Class Model, Physical Component Model and Physical Deployment Model.

As mentioned in Williams (1999), there are many phases in the testing process, including unit, function, system, regression and solution testing. Table 1 shows the differences among these phases, as well as the potential UML diagram used in each phase in terms of coverage criteria and fault model.

Test Phase	Coverage Criteria	Fault Model	UML Diagram
Unit	Code	Correctness, error handling, pre or post condition, invariants	Class and State diagram
Function	Functional	Functional and API behaviour, integration issues	Interaction and Class diagram
System	Operational behaviour	Workload, contention, synchronous, recovery	Use Case, Activity and Interaction Diagram
Regression	Functional	Unexpected behaviour from new or changed function	Interaction and Class diagram
Solution	Inter-system communication	Interoperability problems	Use Case and Deployment Diagram

Table 1: *The Differences Among Testing Phases, Coverage Criteria, Fault Model and Potential UML Diagram to Be Used*

There are a lot of studies about software testing. Offutt and Abdurazik (1999) developed a technique for generating test cases for code from UML state diagrams. They also developed test criteria based on collaboration diagram for static and dynamic testing. The goals of both approaches are to test design models using information from different types of UML diagrams (class and collaboration diagrams). The following part will describe about the implementation of this study.

Implementation

This research was conducted within web-based application environment. The Hotel Reservation System (W-HReS) has been developed in order to test the UML design model. The implementation of this research comprised of activities that are requirements analysis, design the Unified Modeling Language (UML) model, prototype development and test the UML design model.

During this requirements analysis phase, requirements on web-based Hotel Reservation System (W-HReS) for this study were modeled using UML. According to Braude (2000), requirement analysis is a process of understanding and documenting what an application meant to do. The objective of the requirements analysis is to identify what the user

requires from the software elements of the system (Bennett et al., 2002). These requirements have been identified using fact finding techniques such as background reading, interviewing, observation and questionnaires. The output of this phase is a use case diagram.

In UML modeling, use case is the basic diagram to model the requirements. It is a description of set of sequences of actions that a system performs that yields an observable result of value to a particular actor (Booch et al., 1998). A use case diagram represents a set of use cases and actors, and the relationships among them. The W-HReS consists of five use cases which are: `Make Registration`, `Search for Room Availability`, `Make Reservation`, `Login` and `Cancel Reservation`. The use case diagram for W-HReS is depicts in Figure 1.

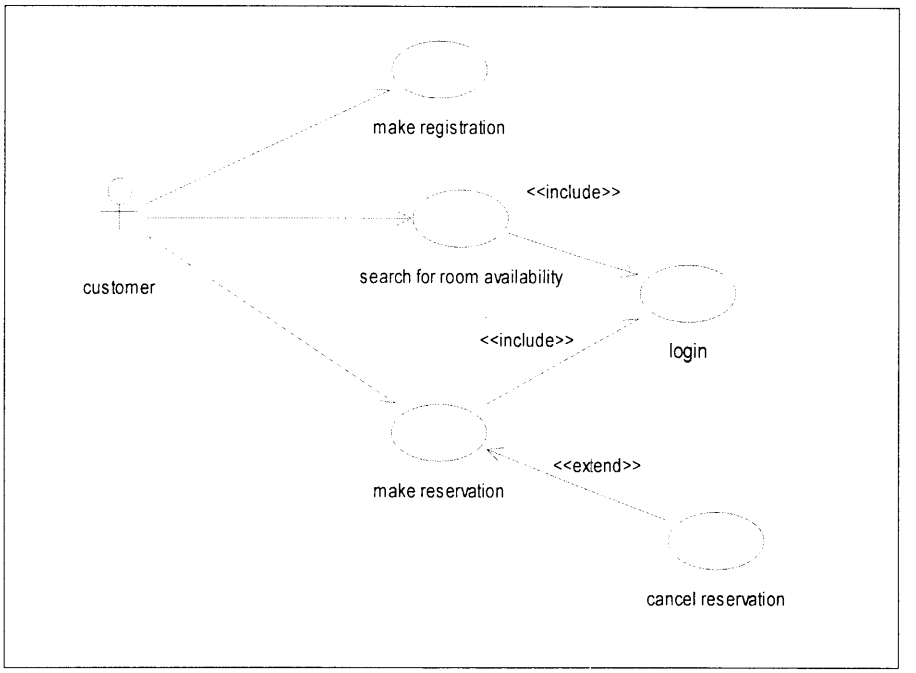

Fig. 1: *Use Case Diagram for W-HReS*

The second activity for this research is designing UML design model. The aim of this phase is to develop Class Diagram for web-based reservation system from the requirements identified in previous phase. For this study, modeling Class Diagram for web-based application proposed by Conallen (1999) has been adopted to model the web design architecture. UML is the standard modeling language for modeling software intensive systems. However, Conallen (1999) claims that the standard UML is not a perfect fit to model all types of applications. Web applications represent one of these applications. Therefore Conallen defines a formal way to extend UML to meet the needs of web-based application. This extension is named UML extension for Web Applications (WAE). In design modeling using WAE, there are three main class stereotypes used in building the class diagram for this study, which are `Server page`,`Client page` and `Form` as illustrated in Table 2. The associations between stereotypes are explained in Table 3.

Name	Description
 Server Page	A server page represents a web page that has scripts that are executed by the server. It also interacts with server-side resources such as databases, business logic components, and external system and so on.
 Client page	An instance of a client page is a Hypertext Markup Language (HTML) formatted web page with a mix of data, presentation and even logic. Client pages are rendered by client browsers and may contain scripts that are interpreted by the browser.
 Form	A class stereotyped as a <<form>> is a collection of input fields that are part of client page. This class maps directly to the HTML <form> tag. Its attributes represent the HTML form's input fields: such as input boxes, text areas, radio buttons, check boxes and hidden fields. A <<form >> has no operations since operation can't be encapsulated in a form.

Table 2: *Class Stereotypes*

Association Stereotypes	Description
`<<link>>`	A `<<link>>` is relationship between a client page and another Web page. In a class diagram a link is an association between a «`client page`» and either another «`client page`» or a «`server page`». A Link association maps directly to the HTML anchor tag, where the `href` attribute is defined.
`<<submit>>`	A «`submit`» association is always between a «`form`» and a «`server page`». Forms *submit* their field values to the server through «`server pages`» for processing. The web server processes the «`server page`», which accepts and uses the information in the submitted form.
`<<build>>`	The «`builds`» relationship is a special relationship that bridges the gap between client and server pages. Server pages only exist on the server. This relationship identifies the HTML output of a server's page execution.
`<<redirect>>`	A «`redirect`» relationship is a directional association with another web page. It can be directed both from and to client and server pages. This association indicates a command to the client to request another resource.

Table 3: *Association Class Stereotypes*

Class diagram for W-HReS has been divided based on the customer roles, registered or unregistered customer. However this paper will only focus on the class diagram for registered user. In this logical view, attributes and operation for each class was identified. Customer has to register first before he/she can search or reserve the hotel room. Customer who is already a registered member can search for available room and make room reservation. In order to login to W-HReS, the customer has to use valid `ID` and `password`. Figure 2 illustrates the detailed class diagram for registered member to search and reserve the room.

A prototype is required to check the user requirements. During this phase, the generic requirements defined in the previous phase will be transformed into codes. In this study, a Web-based Hotel Reservation System or W-HReS was developed as a prototype and used to test the design model. A user can use the W-HReS by using the browser to access

the Hotel Reservation System home page. The home page provides the relevant hotel information. The basic functionality of W-HReS is based on the use case that has been identified before. They are `make registration`, `make reservation`, `login`, `search for room availability` and `cancel reservation`. There are two sections for the user to use the system which are Non-Member and Registered Member section. The user has to register before they can login into the system. Once the user logs in, the user can search for available room and make room reservation. Apart from this, the system also allows the user to cancel their room reservation.

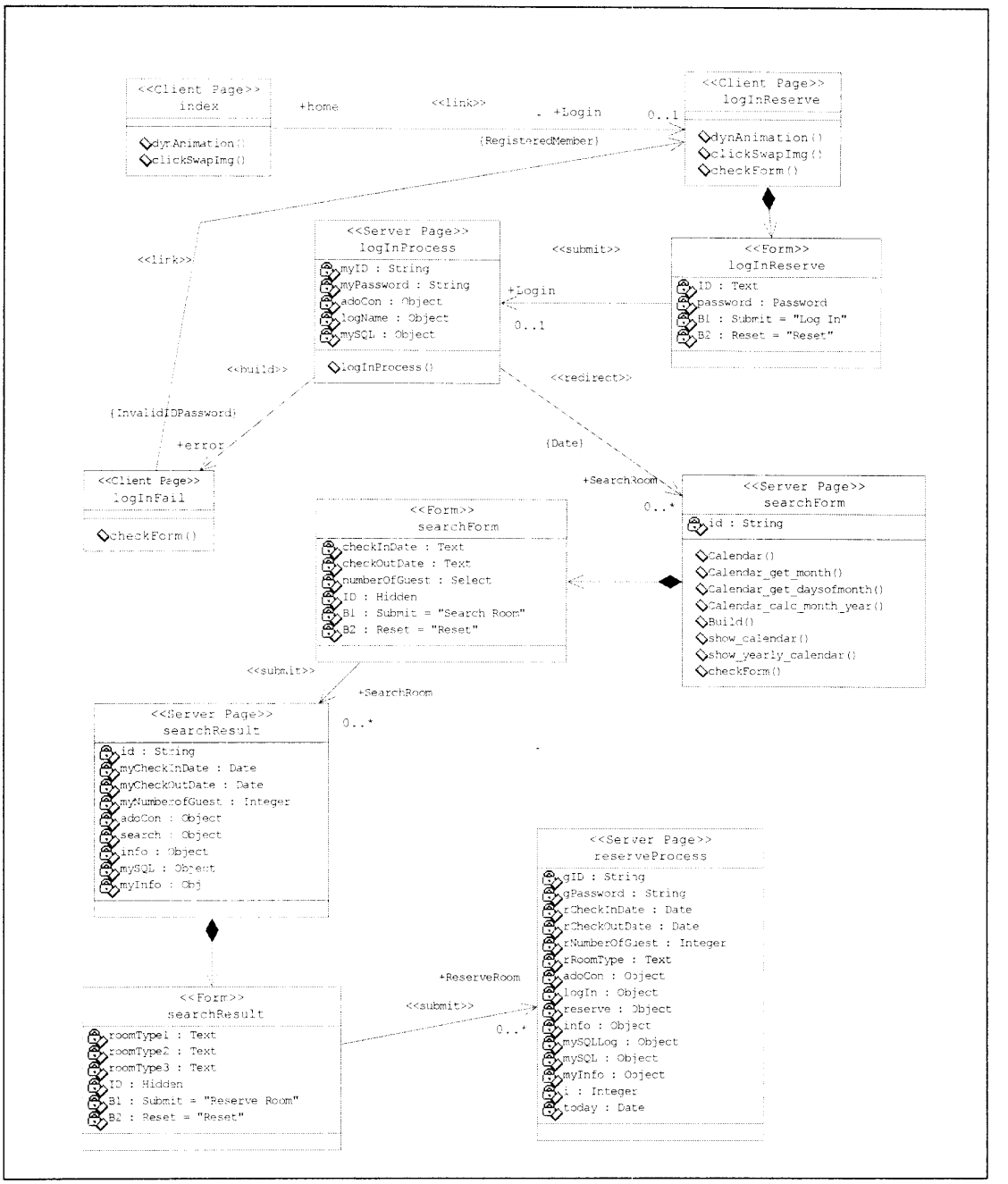

Fig. 2: *Class Diagram for Search and Make Room Reservation*

Testing phase is the most crucial part in validating the design process. Prototype testing is the art of executing software on individuals input values to learn about its behaviour. Testing UML design model has been performed in order to measure and validate the prototype designed. User will input real data and the developer will confirm whether system is reliable or not. The test focuses only on DCD criteria to test and validate the UML Class Diagram for web-based reservation system. Andrews et al. (2003) defined a family of test adequacy criteria for Class Diagrams and Collaboration Diagrams. Test adequacy criteria are set of properties that must be covered during the test. A test is considered adequate if all test adequacy criteria are covered. These criteria are used to guide the selection of test cases and measurement of test adequacy.

Design Class Diagram (DCD) criteria include the following:

- *Association-end Multiplicity (AEM)*: AEM specifies how many instances of a class at the opposite end of association link can be associated with a single instance of a class at the association end.
- *Generalization (GN)*: The GN criterion defines the representatives set of specialization types that must be created from DCD's super classes during the system model test.
- *Class Attribute (CA)*: CA is designed to ensure the testing of behaviours using combinations of representative class attribute values.

In order to establish the set of representative values, a form of category-partitioning technique is adapted. Using this technique, the value domain is partitioned into equivalence classes, which is invalid and valid class. To conduct the testing process, the following assumptions have been made:

1. The class diagram is syntactically correct. This has been ensured by the UML Computer Aided Software Engineering (CASE) tool, Rational Rose.
2. The test case design is performed by using the category partitioning technique. Using this technique the value domain is partitioned into equivalence class, which is valid and invalid class. However the `Button` such as `Submit and Reset` input type do not have the equivalence class. They have only one state.
3. Testing is performed based on the input values from the user. Therefore the test only considered the **<<form>> stereotypes** developed for W-HReS.
4. **Generalization (GN)** criterion was not covered in testing the class diagram since the generalization specialization relationship does not exist in the system.
5. There are **three status** used for testing results, which are:
   - `Not Applicable (NA)`
     NA status is considered when the input entered is invalid and no action done from the pointed class.
   - `Successful (S)`
     S status is considered when the input entered is valid and the system successfully interacts with the associated class.
   - `Unsuccessful (US)`
     US status is considered when the input entered is valid. However the input does not match with the data in the database.
6. **I** and **V** indicate invalid and valid input respectively.

7. **T** stands for `True` and **F** for `False`. `T` indicates that the input parameters can successfully pass the signal to other associated class, and `F` indicates that the input parameters not successfully pass the signal to other associated class.

The implementation of testing process in this paper will only focus on `Search for Room Availability`. This activity or use case requires five main inputs, which are:

- `checkInDate`
- `checkOutDate`
- `numberOfGuest`
- `Submit`
- `Reset`

The tests assume that the user is successfully logged in. The class form for room searching is associated with `searchResult` class. This is shown in Figure 2. Table 4 (Appendix A) shows the test cases derived from DCD for search room availability activity.

Result

As mentioned before, DCD criteria for testing W-HReS class diagram is divided into three criteria; AEM, GN and CA. However, we have found that:

1. Not all criterions in DCD can be applied to test the W-HReS class diagram using WAE. The following explains the findings for each criterion.
   - AEM – Multiplicity in web-based application design model is not apparent. Thus AEM cannot be used to represent all relationships in WAE class diagram.
   - GN – GN criterion is not applied because there is no generalization association in designing web-based application model. So, GN is totally not applicable.
   - CA – CA criterion is apparent and well-defined in WAE class diagram. All attributes are defined in `<<form>>` class stereotypes. Since these criteria can be obtained from these stereotypes, CA is appropriate to test the diagram.
2. A class consists too many inputs will generate complex test cases. It involves a large number of test input parameters combination.

Conclusion and Future Works

This study can be extended to test other UML diagrams such as sequence diagram, collaboration diagram, state chart diagram, sequence diagram and activity diagram for web-based application. In addition the W-HReS can be enhanced using others technologies such as Hypertext Preprocessor (PHP) or Extensible Markup Language (XML) technology. This application can also be integrated with others DBMS like Oracles or MySQL. Besides using the web-based hotel reservation as domain environment, this study can be extended to test applications from other domain such as inventory or procurement management system.

As the conclusion, the designing process of test cases from UML design model for web-based application using DCD criteria was successfully implemented. However, the study found that DCD criteria, in total, are not appropriate for testing web-based application design model. This is due to the fact that AEM cannot be used to represent all relationships in WAE class diagram. In addition, since there is no generalization association in WAE class diagram, GN cannot be used to test the diagram. However CA seems to be appropriate to test the diagram since all criteria are apparent and well-defined in the diagram.

References

1. Andrews, A., France, R., Ghosh, S. & Craig, G. (2003). Test Adequacy Criteria for UML Design Models. *Journal of Software Testing, Verification and Reliability*, 13(2), pp.95–127.
2. Bennett, S., McRobb, S. & Farmer, R. (2002). *Object-Oriented Systems Analysis and Design 2nd Edition*. United Kingdom: McGrawHill.
3. Booch, G., Jacobson, I. & Rumbaugh, J. (1998). *The Unified Software Development Process*. Massachusetts: Addison Wesley.
4. Braude, E.J. (2000). *Software Engineering: An Object-Oriented Perspective*. United States of America: John Wiley & Sons, Inc.
5. Conallen, J. (1999). Modeling Web Application Architectures with UML. *Comm. of the ACM, vol. 42, no. 10*, pp. 63-70.
6. Eriksson, H., & Penker, M. (1998). *UML Toolkit*. United States of America: John-Wiley & Sons, Inc.
7. Jia, X. and Liu, H. (2002). *Rigorous and Automatic Testing of Web Application*. Retrieved April 28, 2004 from http://citeseer.ist.psu.edu/552920.html.
8. Offutt, J., & Abdurazik, A.(1999) Generating test from UML specifications. In *2nd International Conference on the UML*, pp. 416–429.
9. Sparxsystems's UML Tutorial Page (n.d.). Retrieved May 3, 2004, from: http://www.sparxsystems.com.au/UML_Tutorial.htm.
10. Williams, E., C.(1999). *Software Testing and the UML*. Retrieved 20 March, 2004 from: www.chillarege.com/fastabstracts/issre99/99119.pdf

APPENDIX A

No	Test Case Parameters					Associated Class	Coverage Element	
	checkInDate	checkOutDate	numberOfGuest	Submit (Search Room)	Reset	searchResult	Status	Criterion
1	I_ checkInDate	I_ checkOutDate	V_ numberOfGuest	click	-	F	NA	AEM: (1,0): `User (1) -Search Room(0)` CA: `checkInDate, checkOutDate, numberOfGuest, Submit`
2	I_ checkInDate	V_ checkOutDate	V_ numberOfGuest	click	-	F	NA	AEM: (1,0): `User (1) -Search Room(0)` CA: `checkInDate, checkOutDate, numberOfGuest, Submit`
3	V_ checkInDate	I_ checkOutDate	V_ numberOfGuest	click	-	F	NA	AEM: (1,0): `User (1) -Search Room(0)` CA: `checkInDate, checkOutDate, numberOfGuest, Submit`
4	V_ checkInDate	V_ checkOutDate	V_ numberOfGuest	click	-	T	S	AEM: (1,1): `User (1) -Search Room(1)` CA: `checkInDate, checkOutDate, numberOfGuest, Submit`
5	$V_{2\_}$ checkInDate	$V_{2\_}$ checkOutDate	$V_{2\_}$ numberOfGuest	click	-	T	S	AEM: (1,2): `User (1) -Search Room(2)` CA: `checkInDate, checkOutDate, numberOfGuest, Submit`
Click Reset Button								
6	V_ checkInDate	V_ checkOutDate	V_ numberOfGuest	-	click	F	F	AEM: (1,0): `User (1) -Search Room(0)` CA: `checkInDate, checkOutDate, numberOfGuest, Reset`

Table 4: *Test Cases for Search for Room Availability*