An Ontology-based Approach for GUI Testing

Han Li[1,2], Feng Chen[2], Hongji Yang[2], He Guo[1], William Cheng-Chung Chu[3] and Yuansheng Yang[1]

[1] School of Electronic and Information Engineering, Dalian University of Technology, 116023, Dalian, China.
[2] Software Technology Research Laboratory, De Montfort University, Leicester, LE1 9BH, England.
[3] Department of Computer Science and Information Engineering, Tunghai University, Taiwan.

Abstract

As Graphical User Interfaces (GUIs) have almost become ubiquitous for users to interacting with software system, GUI testing becomes an essential task. GUI testing, whose basic steps are test case generation and execution result validation, is a knowledge intensive process that requires both knowledge of GUI systems and testers' experience. In this paper, an ontology-based approach is proposed to make test case generation much effective by involving testers' experience. The approach first establishes a GUI testing ontology by analysing source code with reverse engineering techniques. Next test case generation rules which used to generate test cases are extracted from testers' experience. Then the proposed approach is evaluated. Finally conclusions are drawn and further research directions are speculated.

Keywords: *GUI Testing, Ontology, Test Case Generation Rules, Reverse Engineering*

1. Introduction

Graphical User Interface (GUI) testing is a process of software product testing based on graphical user interfaces [4]. In GUI testing, inadequate testing caused by manual testing and combination explosion of test cases using automatic testing are key issues to be resolved [2]. The main reasons for these issues are the knowledge intensive characters of GUI testing, which include complicated and numerous kinds of GUI components and operations, complex relationships among them and abundant testers' experience. Therefore GUI testing is knowledge intensive, and a knowledge based technique is supposed to be proposed to manage GUI testing.

Based on the advantages of ontology in knowledge representation and reasoning [1, 5], an ontology-based approach to facilitating GUI testing is proposed in this paper. All kinds of GUI elements are described in a GUI ontology, and rules of test case generation are generated by analysing the GUI ontology and testers' experience. Then test cases are generated according to the rules.

2. GUI Ontology

Ontology is able to be used for constructing computable models for some domain [7]. In this paper, for the purpose of using the knowledge provided by a GUI system, GUI ontology is constructed. As shown in Table 1, GUI ontology applied in this paper is composed of a set of concepts, instances, relations and properties, and it is organised in an inheritance hierarchy.

Table 1. Elements and Definitions of GUI Ontology

Element	Definition
Concept	Basic elements in source code
Instance	Implementations of concepts
Relation	Relationships among concepts or instances
Property	Relationships between concepts/instances and data elements

Then according to the design of GUI ontology, reverse engineering techniques [3, 6] are used to capture all elements from source code, and stored them into GUI ontology.

Table 2. User Related Elements and Definitions

Element	Definition	Example
Interactive Component	Concepts that are visible and operable to end users	ListBox …
Database	Database elements	Table…
File	Files except Database elements	.xml, .txt …
Variable	User defined variables used to transfer values	username …

3. Test Case Generation Rules

As test cases of GUI testing are primarily composed of interactive elements and user related data elements, rules of test case generation ought to focus on those user related elements. Thus user related elements are first defined in GUI ontology and classified into

0730-3157/09 $25.00 © 2009 IEEE
DOI 10.1109/COMPSAC.2009.92

different kinds, and then rules are extracted by analysing testers' experience.

3.1. User Related Element Classification

Being a user, except for interactive components such as Buttion, TextBox and Form, all data elements are significant. As the most common data elements in a GUI system are databases, files and variables, user related element are classified into four kinds which are listed in Table 2.

3.2. Rule Generation

Testers' experience refers to GUI component sequences that entered by testers. Different testers have different customs. So in order to extract universal test case generation rules, a large deal of testers' experience is required to be collected.

Based on the knowledge stored in GUI ontology, rules are generated by analysing testers' GUI component sequences. The basic process of rule generation is described as follows.

Step1: Testers' experience is recorded and represented as a large numbers of component sequence in an xml file, where components are those instances defined in GUI ontology. Step2: All components that related to data elements are identified and marked in the sequences. Step3: For each data element, component sequences that related to the data element are collected into the same sequence set. Step4: For each sequence set, the logical relations among components in every sequence is analysed and recorded, where logical relations refer to data dependent or component invoking relations among user related elements. Step5: For each kind of data elements, consequences with similar logical relations are found out, and if the frequency of a logical relation exceeds the average level, it is conclude as a rule for test case generation.

Four main kinds of rules extracted in this research are DB_Rules, File_Rules, Var_rules and Com_Rules. Take DB_Rules as an example, it refers to database related generation rules. The most important DB_Rules is arbitrary combinations of data searching, adding, updating and deleting are needed to be tested. Besides, for the purpose of extending the integrality of test case generation rules, it is possible for testers to add new rules.

4. Result Analysis

First the construction of GUI ontology and the extraction of test case generation rules are semi-automatic.

Then as reverse engineering technology used in this research requires source code, a general business application which is developed for a communication company by our group is chosen to be the case study. In the case study, all 48 interactive components are captured and 12 rules are extracted. Also test cases are able to be generated according to the test case generation rules. Besides, the more testers' experience is provided, the more test case generation rules are extracted.

5. Conclusions

In this paper an ontology-based approach for GUI testing is proposed for the purpose of making use of the knowledge provided by GUI systems and testers' experience. GUI ontology is used to store potential information in a GUI system, while test case generation rules extract useful information from testers' experience. In a word, ontology based GUI testing is a new branch of software testing, which not only takes the knowledge intensive features of GUI testing into account, but also sufficiently make use of them. However, in order to improve the proposed approach, more GUI testing experience is required to be introduced for the purpose of generate more explicit and complete rules of test cast generation.

References

[1] X. Bai, "Ontology-Based Test Modeling and Partition Testing of Web Services," IEEE International Conference on Web Services (ICWS'08), Beijing, China, 2008.

[2] N. Kosindrdecha, "Reducing Test Case Created by Path Oriented Test Case Generation," AIAA Infotech@Aerospace 2007 Conference and Exhibit, California, USA, 2007.

[3] Y. Li, H. Yang, and W. C. Chu, "Fusing ambiguous domain knowledge slices in a reverse engineering process," 7th Asia-Pacific Software Engineering Conference (APSEC'00), Singapore, 2000.

[4] A. M. Memon, "An Event-Flow Model of GUI-based Applications for Testing," *Software Testing, Verification and Reliability*, vol. 17(3), pp. 137-157, 2007.

[5] C. D. Nguyen, "Ontology-based Test Generation for Multiagent Systems," 7th International Conference on Autonomous Agents and Multiagent Systems (ICMAS'08), Estoril, Portugal, 2008.

[6] H. Yang, Z. Cui, and P. O'Brien, "Extracting Ontologies from Legacy Systems for Understanding and Re-Engineering," 23rd Annual International Computer Software and Applications Conference (COMPSAC'99), Phoenix, AZ, 1999.

[7] H. Zhou, J. Kang, F. Chen, and H. Yang, "OPTIMA: an Ontology-based PlaTform-specIfic software Migration Approach," 7th International Conference on Quality Software (QSIC'07), Portland, Oregon, USA, Oct. 2007.