Evaluating the Consistency between Navigation and Data Models for Web Applications

Jordi Cabot, INRIA - Ecole des Mines de Nantes, France
Cristina Gómez, Universitat Politècnica de Catalunya, Spain

ABSTRACT

Specification of web applications involves the definition of two main viewpoints: the navigation model and the data model. Thus far, correctness analysis of the relationship between both models has received little attention. In this paper, the authors present a set of correctness properties to evaluate the consistency of a navigation model with respect to its corresponding data model. Mainly, this paper focuses on the completeness of the navigation model (i.e., if users can manipulate all of the data through the operations in the navigation model) and its executability (i.e., whether all navigation paths admit at least one possible run-time execution that leaves the application data in a consistent state).

Keywords: Data Model, Model Consistency, Navigation Model, Software Quality, Web Engineering

INTRODUCTION

Web development methods have evolved to cover the definition of full-fledged web applications, including data processing and manipulation functionalities. As a consequence, the software models involved in the specification of web applications (i.e., the data model to specify the data manipulated by the application, the navigation model to describe the organization of the front-end interface and the presentation model to personalize its graphical aspect) have been extended with new modeling primitives.

One of the most relevant evolutions is the extension of navigation models with content-modification primitives to define how the user interaction with the pages and links of the navigation model modifies the underlying data of the application. These primitives may be basic CRUD (create/read/update/delete) operations, as in WebML (Ceri et al., 2002) or references to more complex operations (Pastor, Fons, Pelechano, & Abrahão, 2006) (Jakob, Schwarz, Kaiser, & Mitschang, 2006).

These new primitives complicate the correct specification of web applications. Even for small applications, data and navigation models usually turn out to be huge and complex, and, as a consequence, their definition is time consuming and error prone. Poor quality of specifications

DOI: 10.4018/jismd.2010070104
threatens the quality of the final web application, since high-level specifications are usually used to derive the implementation of the web application (e.g., (Ceri et al., 2002), (Pastor, Fons, Pelechano, & Abrahão, 2006), (Melià & Gómez, 2006), (Distante, Pedone, Rossi, & Canfora, 2007) and (Kraus, Knapp, & Koch, 2007) to name a few).

Current proposals devoted to the quality evaluation of web applications have mainly focused on the analysis of the navigation models’ structure (Alalfi, Cordy, & Dean, 2007) as its reachability or the absence of broken links. However, other interesting properties, like those relating the information in the navigation model with the data model are not considered. Indeed, we believe designers would be interested in detecting whether the navigation model is complete, executable and non-redundant as defined in this paper.

A navigation model $N$ is complete with respect to a data model $D$ if users can access and manipulate the values and population of all types in $D$ by means of the read and modification operations appearing in $N$. It is executable when all navigation paths in $N$ admit at least one possible run (i.e., run-time execution) that leaves the underlying data in a consistent state. It is non-redundant when there is a single navigation path to perform each functionality. As an example, consider the simple data and navigation models shown in Figure 1. Clearly, this navigation model is incomplete since it allows users to create new sales but not new sale lines (nor update/delete either sales or lines). It is not executable either. Since sales must be contain at least a sale line, every time the user tries to create a new sale $s$ by using the NewSale page, the execution will fail since the system state is left in an inconsistent state in which $s$ violates the minimum multiplicity of the ComposedOf relationship type.

In this sense, the main goal of this paper is to complement existing correctness properties for web applications with these three new properties focusing on the consistent relationship between navigation and data models. The properties are formalized using a graph-based representation of the navigation model and checked using static analysis techniques that avoid typical efficiency problems (e.g., state-explosion problem (Clarke, Grumberg, Jha, & Lu, 2001)) of simulation-based techniques. Moreover, as a result, our analysis provides designers with valuable feedback that help them to repair possible errors in the specification.

This paper extends our previous work (Cabot & Gómez, 2007) in several directions. First, the list of correctness properties has been refined and extended. Second, a basic set of graph-based algorithms to verify each property is provided (including the possibility of incrementally verifying the properties after changes on already consistent versions of the models). A more complete related work is also provided.

The rest of the paper is structured as follows. In the next section, we review some basic concepts of data and navigation models. Next, we present our graph-based representation for navigation models. In “Completeness”, “Executability” and “Non-Redundancy” sections, we formalize our quality properties. “Incremental Consistency Checking” addresses the efficient verification of the properties. Section “Tool
Related Content

Multidimensional Ontology-Based Information Retrieval for Academic Counseling
[www.igi-global.com/article/multidimensional-ontology-based-information-retrieval-for-academic-counseling/117769?camid=4v1a](www.igi-global.com/article/multidimensional-ontology-based-information-retrieval-for-academic-counseling/117769?camid=4v1a)

An Alternative Threat Model-based Approach for Security Testing
[www.igi-global.com/article/an-alternative-threat-model-based-approach-for-security-testing/136452?camid=4v1a](www.igi-global.com/article/an-alternative-threat-model-based-approach-for-security-testing/136452?camid=4v1a)

Fault-Prone Module Prediction Approaches Using Identifiers in Source Code
[www.igi-global.com/article/fault-prone-module-prediction-approaches-using-identifiers-in-source-code/121546?camid=4v1a](www.igi-global.com/article/fault-prone-module-prediction-approaches-using-identifiers-in-source-code/121546?camid=4v1a)
Usability Engineering Methods and Tools
www.igi-global.com/chapter/usability-engineering-methods-tools/74882?camid=4v1a