Chapter V

Anomaly Detection and Quality Evaluation of Web Applications

May Haydar
Université de Montréal, Canada

Ghazwa Malak
Université de Montréal, Canada

Houari Sahraoui
Université de Montréal, Canada

Alexandre Petrenko
Centre de recherche informatique de Montréal (CRIM), Canada

Sergiy Boroday
Centre de recherche informatique de Montréal (CRIM), Canada

ABSTRACT

This chapter addresses the problem of Web application quality assessment from two perspectives. First, it shows the use of model checking of properties formulated in LTL to detect anomalies in Web applications. Anomalies can be derived from standard quality principles or defined for a specific organization or application. The detection is performed on communicating automata models inferred from execution traces. Second, the chapter explains how probabilistic models (Bayesian networks) can be built and used to evaluate quality characteristics. The structure of the networks is defined by refinement of existing models, where the parameters (probabilities and probability tables) are set using expert judgment and fuzzy clustering of empirical data. The two proposed approaches are evaluated and a discussion on how they complement each other is presented.
INTRODUCTION

The Internet has reshaped the way people deal with information. A few years ago, simple Web sites existed, where the components were text documents interconnected through hyperlinks. Nowadays, the Internet and the Web affect daily life in many ways. They are used to run large-scale software applications relating to almost all aspects of life, including information management/gathering, information distribution, e-commerce (business-to-customer, business to business), software development, learning, education, collaborative work, and so forth. According to Offut (2002), diversity is a key description of Web applications (WA) in many aspects that led to the notion of “Web engineering.” Web applications are developed with cutting edge technologies and interact with users, databases, and other applications. They also use software components that could be geographically distributed and communicate through different media. Web applications are constructed of many heterogeneous components, including plain HTML files, mixtures of HTML, XML, and programs, scripting languages (CGI, ASP, JSP, PHP, servlets, etc.), databases, graphical images, and complex user interfaces. These diversities led to the need for large teams of Web developers who do not share the same skills, experience, and knowledge. These include programmers, usability engineers, data communications and network experts, database administrators, information layout specialists, and graphic designers [38]. With such a diversity of Web applications developers, quality is a primary concern. Unlike traditional software, Web applications have an extremely short development and evolution life cycle and have to meet stringent time to market requirements. Web applications often have a large number of untrained users, who often experiment with the Web applications unpredictably. The success of Web applications solely depends on their users and their satisfaction. Hence, a low quality of these applications can be very costly; as an example, the 4-day outage of Microsoft Money in 2004 was caused by a server glitch that prevented users from accessing their online personal finance files (Pertet & Narasimhan, 2005). Microsoft Money’s servers were unable to recognize usernames and passwords through the Microsoft’s Passport authentication and log-in service. Therefore, thorough analysis and verification of WA is indispensable to assure their high quality.

There exist at least two different perspectives for dealing with quality of Web applications. The first one concentrates on detecting and correcting anomalies and the second viewpoint focuses on building their quality models. In this chapter, two approaches are described, and both perspectives are elaborated, namely detecting anomalies using model checking of execution traces and quality evaluation using probabilistic quality models.

One approach uses formal methods for the analysis and validation of Web applications. The idea is to observe the executions of a given Web application from which its automata-based models are inferred. Using existing quality and usability rules that assess Web application’s design and implementation, properties in linear temporal logic (LTL) are formulated; for more details on LTL, see Clarke, Grumberg, and Peled (2000). The model and properties are fed to the model checker Spin that verifies if the model satisfies those properties. The model checker then provides a counter example in case the property is not satisfied in the model. Counter example information helps in the evaluation and correction of Web applications.

Another approach relies on a probabilistic quality model involving Bayesian networks and fuzzy logic to assess the quality of Web applications. Web applications quality criteria, proposed in the literature Olsina (1998), Nielsen (2000), and Koyani, Bailey, and Nall (2003), are collected along with a list of the existing guidelines and recommendations. The list is extended by considering additional criteria that are also significant.