# Categorization of Web Application Errors

By Libby Johnson
August 2005

According to L. Xu, B. Xu, and J. Jiang, the unique features and complexity of web applications make it difficult to apply traditional testing theories and methods.[1] I assert that the same is true for traditional categorizations of errors. Characteristics that make web applications different from traditional applications are their distributed, multi-platform, and dynamic natures as well as use of hypertext and hypermedia.[1] These unique characteristics will make for unique errors.

In my research, I could find no framework specific for classifying and discussing errors in web applications. Work on web application analysis has concentrated on defining testing methods and criteria. Although this work focuses on testing, it may aid in categorizing web application errors. Since testing seeks to systematically expose errors, it follows that logical categories for testing represent logical categories of errors.

I propose a framework for classifying errors in web applications based on analysis and logical categorization of web application elements developed for testing. Figure 1 graphically represents this framework.

The framework moves from general to specific in its classification of errors and first classifies errors as either static or dynamic. This division is adopted from those made by Sciasco, Donini, Mongiello, and Piscitelli and Ricca and Tonella to develop testing criteria. Sciasco et al define static web pages as those containing fixed content including pages with client-side script. They define dynamic pages as those that generate content at run time and therefore include server-side script.[2] Earlier work by Ricca and Tonella also classify web pages as either dynamic or static where, dynamic pages are those whose content is computed at runtime.[3] Therefore, static errors may be defined as those present in a web page feature defined before runtime. Likewise, dynamic errors are those present in web page features generated at runtime.

Under the static and dynamic descriptors, errors are further subdivided into functional errors and usability errors. L. Xu, B. Xu, and J. Jiang apply two familiar concepts of testing to web applications – functionality and usability. Functionality testing aims to ensure functional correctness according to specifications. Usability testing, on the other hand, tries to ensure that features are consistent, concise, clear, and have well defined purpose.[1] For classification purposes, functional errors can be described as errors that cause the site not to behave as specified or intended. Usability errors are errors that inhibit a user's ability to efficiently use a web site. Functional and usability errors can appear in static and dynamic web pages and can be applied to further categorize static and dynamic errors.

The functional and usability error types will first be examined as they apply to static errors. The category of functional, static errors subdivides further into categories labeled syntax and navigation. This is based on Xu, Xu, and Jiang's work which tests HTML and hyperlinks as components of functionality.[1] Examples of syntax errors in web pages are errors in html or client-side scripts. Hyperlinks, of course, comprise navigation. Html, scripts, and hyperlinks clearly contribute to how a site functions consistent with our earlier definition of functionality. In addition, these elements are static because they define web page elements prior to runtime.

Usability errors are a result of design errors. Xu, Xu, and Jiang examine graphics and appearance in their usability testing.[1] Newman and Landay consider content, color, images, typography, and layout elements of web site design. These design elements affect the communication of information in a site.[4] Navigation plays a roll in a website's usability as well because, in addition to providing the functionality websites are prized for, it must be consistent, concise, clear, and have well defined purpose – in other words, easy to use.[1] Form design and layout can be considered static. The data collection features of forms are dynamic and will be discussed under the dynamic, functional, run time categories.[3]

Functional and usability categorizations apply to dynamic errors as well. Functional errors describe a deviation in expected behavior and dynamic errors describe errors in features generated at run time. Therefore, dynamic, functional errors are more specifically termed run time errors. The primary concerns for dynamic pages in Sciasco et al's model are security and access control.[2] These features determine how secure a web site's information is and who can access it. How forms function in their submission and retrieval of data is also a run time concern. When data is collected it is, "submitted to the Web server via the special link *submit*, whose target is always a dynamic page."[3] Forms can also display data generated at run time, such as a table of results from a dynamic query of a database. Errors in this process can be termed run time errors. Databases and forms are related because the two generally interact in web applications. Submission to and retrieval of information in a database are dynamic features invoked at run time. Finally, performance is a somewhat fuzzy feature but, I have chosen to put it in the run time category under dynamic, functional errors. Xu, Xu, and Jiang give performance its own category in their testing schema. However, I choose to classify performance errors in the dynamic, functional, run time categories. Performance encompasses site response time and user load handling capabilities. An example of a performance issue is customers leaving a site because of slow response times or unavailability.[1] Although, this seems like a usability issue – customers leaving a site because it is hard to use, I propose that it is a functionality issue – customers leaving a site because it does not function properly. This latter view assumes that reasonable response time and availability are functional requirements for web applications.

There can also be usability errors in dynamic web page components. Dynamic usability errors, like static usability errors, result from flaws in design. Examples of design elements for dynamic features are data validation, defaults, and history (remembering fields previously filled in) for forms. The capability to dynamically query databases and ability to add and remove fields in forms and databases are other design elements. The design aspect of dynamic features can have substantial affect on the usability of a web site. For example, suppose a user is filling out a complex form and chooses to move to the next page but has forgotten to fill in a required field. It is common practice for the user to be instructed to fill in the required field and redirected back to the form page. It is quite annoying to have to fill in the entire form again. However if all the form data is saved, the user simply enters the missing information and moves on, making the web site much more usable.

In summary, web site development has evolved from an individual's creative pursuit into the engineering of large scale applications. Sciasco, Donini, Mongiello, and Piscitelli remind us that, as a new discipline, Web Engineering needs a formal framework for defining the quality of web applications.[2] As part of that framework, a formal system of categorizing errors would help researchers and testers develop methodologies for identifying and predicting errors. The classification framework in this paper attempts to categorize the types of errors that occur in web applications. Five specific categories of errors have been described: syntax, navigation, static design, run time, and dynamic design. This classification framework acknowledges that functionality and usability are intertwined in a successful web application. The framework also accounts for the inclusion of both static and dynamic features in web sites. While this classification framework attempts to be thorough, no doubt more work will need to be done in this area.

## Definitions

Static error – error present in a web page feature that is defined before run time

Dynamic error – error present in a web page feature generated at run time

Functional error – error that causes a web application not to behave as specified or intended

Usability error – error that inhibits a user's ability to efficiently use a web site

## References

[1] Lei Xu, Baowen Xu, Jixiang Jiang, "Testing Web Applications Focusing on Their Specialties," ACM SIGSOFT Software Engineering Notes, Vol. 30 Number 1, January 2005, p. 1, 3, 4.

[2] Eugenio Di Sciascio, Francesco Donini, Marina Mongiello, Giacomo Piscitelli, "AnWeb: a System for Automatic Support to Web Application Verification," Proc. ACM Conf. Software Engineering and Knowledge Engineering, July 2002, p. 609, 612.

[3] F. Ricca, P. Tonella, "Analysis and Testing of Web Applications," Proc. ACM Conf. Software Engineering, July 2001, p. 26, 28.

[4] Mark W. Newman and James A. Landay, "Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice," Proc. ACM Conf. Designing Interactive Systems: Processes, Practices, Methods, and Techniques, August 2000, p. 264.