Smart Web Forms

A Europython 2004 presentation
June 7th to June 9th 2004

Fotios Basagiannis
Joint Research Centre
European Commission

Abstract

Complex web applications collect complex data using inevitably complex web forms that give rise to user cognitive overload. Using python in the back end and javascript in the front end, the smart forms framework generates interactive web forms, out of xml based descriptions, that minimize complexity by presenting the form fields hierarchically (with nested elements being collapsed). The xml describes the form elements, their presentation semantics and declares the DOM events that trigger their interactive behavior. The smart forms framework simplifies complex forms in an intuitive and effective manner.

Introduction

This is an xml based framework for generating interactive web forms using python and zope

Complex web applications collect complex data using inevitably complex web forms that give rise to user cognitive overload.

Web user cognitive overload has been shown to:

- Decrease the quality of user input
- Decrease the number of users that go though complex forms

This complexity is usually dealt with in one of the following manners:

- Long or complex forms are forcibly split into multiple sub-pages.
- Server side logic dynamically builds customized versions of the forms based on user interaction.
- Client side logic dymamically builds/alters the forms based on user interaction

The first method has the following problems:

- The resulting multi-page form can act as a deterrent to people who are not very motivated to fill out a partcular form.
- Conceptually related information is forcibly segmented making user references to previously entered data difficult.

The second method has the following problems:

- Slow responses to user interaction in heavy network traffic situations or when the user is behind a slow modem.
- Added processing load on the back end servers.

The third method has the following problems:

- Javascript code tends to be buggy and not cross-browser enough
- Solutions of the kind tend to be application specific and do not provide a robust cross-browser framework - built around an intuitive enough abstraction - that is reusable.

Any of these methods may (or may not) be using xml in the back end (or even the front end) to automate form building using some web-form describing xml schema.

The framework that this paper presents uses the third method and attempts to completely alleviate problem 2 of it while offering a javascript implementation that is compatible with the following browsers:

- IE 5.0+
- Gecko based browsers (i.e. Mozilla 1.0+)
- Opera 7.0+ (untested - possibly needs some extra work)

The paper also defines an xml schema for specifying interactive (smart) forms, i.e. a schema that allows not only for the description of the structure of a form but also for the description of interaction semantics between user and form (e.g. show these extra form elements if the user clicks on this radio button)

The presented framework tackles the form complexity problem using a folding metaphor whereby the form is divided into foldable sections that only show when necessary (and according to the interaction specification in the form's xml)

The framework allows for the use of XSL in the the presentation of the form (but presentation may also be described using a separate namespace in the form's xml)

Our implementation of the framework uses python/zope on the back-end and JavaScript on the front end. <eXpat/> does the xml parsing and we are thinking of using 4suite for our XSL needs.