

# CocoonCFormsXULUIProject

First specification for the [XUL based user interface for Cocoon forms project](#) of the [GoogleSummerOfCode2005](#).

The project mentors are [AntonioGallardo] and [JoergHeinicke].

## Project Title

XUL Based User Interface for Apache Cocoon Forms

## Synopsis

Create a new user interface rendering for cforms using Mozilla XUL, The user interface should be easily switchable HTML-XUL depending on the client browser.

## Background/Motivation

Apache Cocoon is a Web application development framework built around the concepts of XML processing pipelines and component-based application development, it takes Model-View-Controller distinctions and adds an emphasis on Separation of Concerns (SoC), which makes it like the "web glue" that hooks together components into pipelines and allows parallel evolution of all aspects of a web application, hence improves development pace and reduces the chance of conflicts. The project was started in 1999 and has since then grown both in features and user base, and now is a top-level project of the Apache Software Foundation, and has been widely used by a lot of people and organizations around the world for developing mission-critical web applications.

Cocoon Forms, or simply referred to as CForms, is Cocoon's advanced XML form framework. Combined with the powerful Flow mechanisms which use the Continuations programming paradigm, and the JXTemplate generator, CForms is potentially revolutionary for web application developers, allowing highly interactive applications that run in a standard browser to be developed with unusual speed. CForms is still under development, the developers are working towards its first stable release.

Mozilla XUL(XML User Interface Language) is one component of the Mozilla application development suite, currently being used to develop the integrated Mozilla application, like the Firefox browser, the Thunderbird e-mail client, and so on. The architecture depends upon a core library XPCOM (cross platform COM), with applications developed using XUL to define user interfaces and [JavaScript](#) to control application flow.

Currently CForms only renders HTML-based user interface, with the surging popularity of XUL-supported web browser Firefox, and the increasing interests on developing rich client applications which build upon a well designed framework with lightweight XML and [JavaScript](#) so that new application functionality can be quickly developed and deployed, there is strong need to add XUL support to CForms, thus extend the GUI features and functionalities of CForms and provide a solid basis for creating more interactive rich client applications.

The purpose of this project is therefore to build a set of form widgets, XML styling files, flow control scripts and pipeline code to enable CForms "XUL-aware" and provide transparent XUL support for CForms, also allow CForms developers to easily manage the GUI rendering, and be able to choose between HTML and XML based user interfaces depending on different client browsers.

## Project Description

The project is divided into three parts, each part deals with one aspect of the Model, View and Controller aspects of CForms respectively to enable XUL support. Specifically:

### Part 1 Model

In CForms, the form model is defined by a form definition which is an XML file describing the structure of the form widgets it consists of. A form instance, which will hold actual form data, can be created based on the form definition. Although the user interfaces are defined in XML within XUL framework, the separation between model and view - that is, between the data to be processed and the physical controls on screen to manipulate the data - is not as clearly defined as in CForms. So the Form class in CForms need to be extended to extract the form widgets from XUL files, and be able to hold the form state info from a Request object for every XUL form fields.

### Part 2 View

The view of a form in CForms is handled by a form template which is a XML file specifying where the form widgets to appear by inserting special tags referencing that widget. The form template is processed by a publishing pipeline, which will go through Forms Template Transformer, XSLT, and HTML Serializer, to render plain HTML. Compared to HTML forms, XUL has no equivalent to the <FORM> tag. XUL form control equivalents are not bound into groups (except via <radiogroup>). There is no equivalent to <INPUT TYPE="reset"> or <INPUT TYPE="submit"> in XUL. So to expand the Publishing pipeline to generate XUL forms, we need to update the form XSL files to accommodate the XML representations of XUL widgets, as well as to add new transformers to handle special XUL widget stylings.

### Part 3 Controller

Since XUL has no semiautomatic form submission process, if a plain XUL document is presented to the user, then nothing happens to the data entered by the user. This means that if a XUL application is to behave like an HTML form, then form submission scripts need to be added to each XUL form. So we need to implement XUL form submission functionality in Form class of CForms. The technique we choose to handle XUL form submission is to use XMLHttpRequest object, which allows HTTP requests to be submitted directly from [JavaScript](#). The HTTP request can be submitted synchronously or asynchronously.

Synchronous submission means that the script halts until the full response is received. Asynchronous submission is a "fire and forget" system, except that progress can be tracked, and the final result can be recalled. We choose asynchronous submission in our implementation. After the form is submitted, the controller will let the form instance object process the request, so that all widgets can read their value from the request.

Finally, we'll design and implement a GUIManager module to integrate with the publishing pipeline to handle form user interface issues, depending on different client browsers, the GUIManager should be able to choose between HTML and XML based user interfaces and render the interface accordingly.

## Deliverables

code: one module for XUL Form model, one XUL widget styling file, one XUL widget transformer, one module for XUL form submission, one module for GUIManager.

technical paper: one paper to describe the technical details of the project.

tutorial: one tutorial to demonstrate how to use the XUL related modules.

## Project Schedule

- 06/15/05 - 06/25/05: warm up, get ready on technical side, waiting for the approval of proposal
- 06/26/05 - 07/09/05: XUL form model
- 07/10/05 - 07/16/05: XUL widget styling
- 07/17/05 - 07/23/05: XUL widget transformer
- 07/24/05 - 08/06/05: XUL form submission
- 08/07/05 - 08/20/05: GUIManager
- 08/21/05 - 08/31/05: Wrap up, paper and tutorial

## Participants

- Developer: He Huang, University of Southern California - email: [heh.AT.usc.edu](mailto:heh.AT.usc.edu)
- Mentors: [AntonioGallardo] and [JorgHeinicke] from The Apache Cocoon team

## References

- Apache Cocoon project: <http://cocoon.apache.org/2.1/index.html>
- Apache Cocoon Forms: <http://cocoon.apache.org/2.1/userdocs/forms/>
- Creating Applications with Mozilla: <http://books.mozdev.org/index.html>
- Rapid Application Development with Mozilla: <http://mb.eschew.org/>