CM0343 – Individual Project

Initial Plan

“Configurable User Interfaces with XML”

Author: James Briggs
Supervisor: Frank C Langbein
Moderator: Xianfang Sun

Project Description

This project will aim to create a language and set of associated tools capable of prescribing the layout of Graphical User Interface (GUI) elements/widgets within a standardised XML document. The project draws comparisons to existing systems such as HTML/CSS or XUL as used by the Mozilla Firefox web browser. Currently there are no popular unified formats for specifying a GUI without being tied to a particular programming language. For example, creating a GUI in Java involves creating Swing elements programmatically as opposed to using a separate file defining the elements and layout. My project aims to solve this problem by creating a cross platform, flexible GUI development framework as shown in the diagram below. The project can be split into several different areas, the core being the creation of the XML schema. I will build upon existing technologies such as XML and Document Types Definitions (DTDs) to ensure that layouts conform to a pre-agreed schema. In order to specify a layout two XML files will be created: one will describe the elements within the interface, the other will describe their attributes such as position and dimensions. In order to simplify the creation of these documents I will also implement a visual GUI editor similar to those found in Microsoft Visual Studio or Eclipse. In addition advanced users will still have the option of creating layouts by hand in a text editor. In order to render and manipulate the GUI within a program I will create APIs/libraries with functionality similar to that between JavaScript and the Document Object Model (DOM). One of the major goals will be to ensure that the project is cross platform in regard to both operating system and programming language/GUI toolkit. I will target two different programming languages/GUI toolkits: SWING in Java and wxWidgets in Python. Rather than compiling Java/Python code specifying the GUI the actual XML file will be used in an interpretative manner. This means that layout files can be moved across different platforms more easily without need for recompilation. The project can be used in any context where an application requires a GUI. As a demonstration of the project's functionality and versatility the GUI editor could itself recursively use the XML formats and libraries developed for its layout.

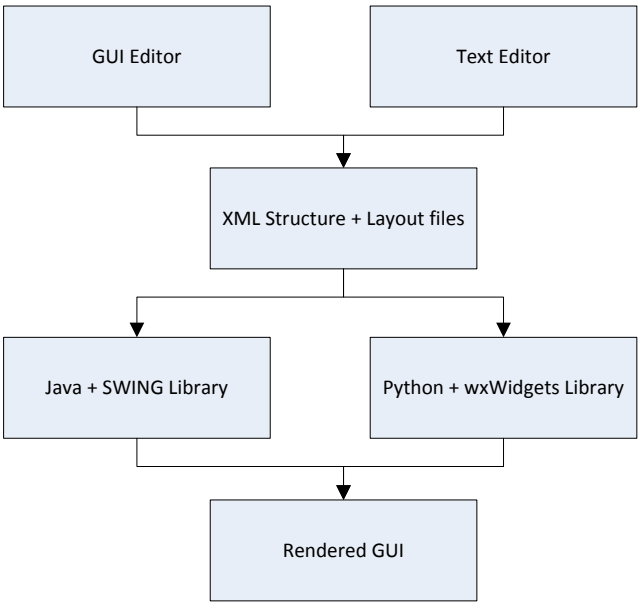

Diagram showing the general architecture of the system from creation of the GUI through to rendering within an application.

Aims and Objectives

- Create an XML schema and associated DTDs that can be used to describe the layout of a GUI.
    - Create a strict yet flexible/expandable standard for writing GUIs in XML.
    - Fully document the schema.
    - Allow similar layouts to be produced on all major desktop platforms (Windows, Mac and Linux).
- API libraries.
    - Produce lightweight API libraries for both Python and Java.
    - Emulate the functionality of the Document Object Model for desktop applications.
    - Allow full control over elements via the API.
    - Allow XML layout and structure files to be loaded and rendered with minimal lines of code.
- Visual GUI editor.
    - Develop a 'drag-and-drop' GUI editor using either Python or Java.
    - Allow novice users to design effective GUIs whilst still giving advanced users the options of writing pure XML.
    - Allow GUI layout/structure files to be easily created, saved and opened for editing.
    - Offer a flexible and cross-platform selection of widgets with easily customisable options such as size and position.
- Example applications.
    - Produce 2-3 applications demonstrating the use of the system.
    - These will be fairly generic applications such as a calculator and GUI interfaces to command line tools.
    - A wide variety of GUI widgets will be used to demonstrate the scope and flexibility of the system.
- Interim report.
    - Include details on the background research including justification for the programming languages, libraries and standards used.
    - Explanation of the need for such a system and what problems it solves.
    - Details of the approach I have taken/plan to take to solve the problem.
    - Examples of the projects progress, possibly a demo application or sample GUI.
- Final report.
    - Design of the overall system.
    - Results of the project including benchmarks and comparisons to existing systems and methods such as hard coding a GUI.
    - Evaluation of the implementation.
    - Potential future expansions including details on the flexible architecture of the system.

Work Plan

I will use the Agile software development model throughout the project as it lends itself well to the requirements and nature of the project. The flexibility of the system means that it is difficult to determine exactly how long certain features may take to implement and whether they can e implemented at all. I feel discussing change and responding to limitations is a more effective strategy than contract negotiation for this project.

Week 1 (Oct 1st 2012)	**Milestones and Deadlines**
Introduction to the project. Some of the aims were discussed in a meeting with supervisor.	-
Week 2 (Oct 8th 2012)	
Write the draft project plan.	Finish draft version of the project plan in time for supervisor/moderator review and submission next week.
Week 3 (Oct 15th 2012)	
Amend plan if necessary and agree it with the supervisor and moderator. Submit final version by 23:00 on Friday. Start Background research. This will involve finding the best way XML can be used within the project including research into Document Type Definitions to control the document structure and create a standardised format. This involves looking into existing systems such as XUL.	**Friday Oct 19th 2012 – Project Plan Deadline.**
Week 4 (Oct 22nd 2012)	
Complete background research. Research suitable XML/DTD parsing libraries for Java and Python as well as which GUI elements I wish to include in the schema. Start implementation.	Brief notes will be created for use during the development.
Week 5 (Oct 29th 2012)	
I will create the DTD specification along with a small sample XML file. I will document each of the elements included in the specification along with each of their attributes.	Complete DTDs, sample XML schema and draft documentation outlining GUI components and their properties.
Week 6 (Nov 5th 2012)	
I will continue to develop the XML schema during this week.	Further draft documentation and XML schema definitions.
Week 7 (Nov 12th 2012)	
I will develop one of the API libraries (Java or Python) during this week so that hand coded XML documents can be loaded, parsed and rendered in an application.	Initial API library.
Week 8 (Nov 19th 2012)	
I will spend this week finalising the previous API and creating the remaining one (Java or Python). I predict it will take less time to develop the second one as the code will be similar (if not syntactically) to the	Initial API libraries for both Java and Python.

previous one. I will spend part of the time starting the interim report.	
Week 9 (Nov 26th 2012)	
I will use the final week of November to develop a small sample application demonstrating the functionality of the project to date. I will continue to work on the interim report.	Demo application.
Week 10 (Dec 3rd 2012)	
I will focus on the interim report during this week so that I can ensure it is of a high quality and iron out any problems well before the deadline.	Draft interim report.
Week 11 (Dec 10th 2012)	
This week will be used to complete and proof read the interim report and agree it with the supervisor.	**Wednesday Dec 14th 2012 – Interim Report Deadline.**
Christmas Recess – Week 1 (Dec 17th 2012)	
Finalise current implementation. This includes brief testing and ironing out major bugs.	Initial API libraries / demo.
Christmas Recess – Week 2 (Dec 24th 2012)	
I will continue to finalise the implementation.	Initial API libraries / demo.
Christmas Recess – Week 3 (Dec 31st 2012)	
I will continue to finalise the implementation however revision for upcoming exams will be the focus.	Final version of the API libraries.
Exams – Week 1 (Jan 7th 2013)	
Revision week.	-
Exams – Week 2 (Jan 14th 2013)	
Revision week.	-
Exams – Week 3 (Jan 21st 2013)	
Revision week.	-
Week 1 (Jan 28th 2013)	
I will spend this time researching existing graphical GUI editors such as Eclipse and Visual studio as well as searching for suitable Python/Java libraries to help with the implementation.	Brief notes on the background research conducted.
Week 2 (Feb 4th 2013)	
This week I will focus on developing the GUI editor. I will decide upon a suitable language (either Java or Python) based upon the research conducted last week. I will most likely create the skeleton of the application (positioning all the elements, setting up basic functionality and menus).	Basic skeleton of the GUI editor.
Week 3 (Feb 11th 2013)	
I will continue development of the GUI editor this week. I aim to finish the interactive widget 'drag-and-drop' functionality.	Enhanced version of the GUI editor.
Week 4 (Feb 18th 2013)	

This will I will finish the final version of the GUI editor including all file handling.	Final version of the GUI editor.
Week 5 (Feb 25th 2013)	
I will rigorously test the API Libraries and GUI editor to ensure all bugs are fixed and the project fulfils all requirements. I will produce draft documentation explaining how to use the system.	Draft documentation. Testing evidence/documentation.
Week 6 (Mar 4th 2013)	
I will create a set of demo applications (2-3) to demonstrate the functionality of the project with a side benefit of testing parts of the GUI editor.	Draft demo applications.
Week 7 (Mar 11th 2013)	
I consider this the final week of the technical side of the project. I will spend this week completing the demo applications which are essentially an evaluation of the project. I will polish all elements of the implementation.	Final demo applications. Polished version of the implementation.
Week 8 (Mar 18th 2013)	
During this week I will complete the documentation for the system describing its high level use and lower level functions.	Documentation.
Easter Recess – Week 1 (Mar 25th 2013)	
Evaluation – I will evaluate the system based on results from the demo applications.	Draft evaluation.
Easter Recess – Week 2 (Apr 1st 2013)	
I will complete the evaluation. Additionally I will start planning the final report.	Final evaluation. Draft final report plan.
Easter Recess – Week 3 (Apr 8th 2013)	
Complete final report plan.	Final report plan.
Week 9 (Apr 15th 2013)	
I will continue with the final report throughout this week.	Draft final report.
Week 10 (Apr 22nd 2013)	
I will continue with the final report throughout this week.	Draft final report.
Week 11 (Apr 29th 2013)	
I will finish and proof read the final report during this week as well as agreeing it with the supervisor and moderator.	**Friday May 3rd 2013 – Final Report Deadline.**