

## XFlash – A Web Application Design Framework with Model-Driven Methodology

Ronnie Cheung  
Hong Kong Polytechnic University,  
Hong Kong SAR,  
China  
[csonnie@gmail.com](mailto:csonnie@gmail.com)

**Abstract.** Developing web-based applications is a time-consuming task. Because of the diversity and complexity of web applications, it is difficult to develop web applications in an object-oriented manner. In terms of the Model-View-Control framework (MVC), the control layer has two responsibilities, one is to retrieve the data for the view layer, and the other is to control the navigational structure of the view layer. This makes the boundary between the control layer and view layer unclear. To address these issues, we have developed a web application design framework called XFlash for generating reusable web-based components. In the XFlash framework, we have applied the object-oriented hypermedia design methodology to model web application structures, and provide a model-driven approach for the view layer design. By using this approach, it is possible to develop web user-interfaces in a model-driven manner.

**Keywords:** Object-oriented hypermedia design methodology, Web application design framework

### 1 Introduction

Developing a web-based application is a time-consuming task. Because of the diversity and complexity of web applications, it is difficult to develop a web application in an object-oriented manner. Web engineering differs from the traditional software engineering approaches for software development. The design of web applications must be flexible enough to support adaptive changes. Components provide maximum flexibility to increase their reusability in different applications and to work under changing environments [5]. However, traditional component-based software engineering approaches are not suitable for web-based application development. The developer of web applications needs to implement the system with several languages, such as HTML, JavaScript, and Java. It is difficult for the developer to extract the web user-interface component models for reuse.

In the XFlash framework, we provide a model-driven approach for web user-interface development. The object-oriented hypermedia design methodology (OOHDM) [2] is used to model the web application structure. By providing enhancements to the OOHDM approach, we design the navigational structure of a

web application using the abstract user-interfaces model rather than the navigational model. In terms of the abstract user-interfaces model, we applied the component-based model driven approach [4] to designing the web user-interfaces.

In order to validate our approach, we implemented a model driven-framework - XFlash. The XFlash framework is based on the OOHDM approach. It helps developers to design the user-interfaces model with a visual graphical editor. For traditional web applications, web user-interface components are built out of several language, such as HTML, JavaScript and Java. It is difficult for developers to implement web user-interface components in an object-oriented manner. In the XFlash framework, web user-interfaces components are implemented with a single language - ActionScript. Furthermore, the XFlash framework generates flash movies to build the web user-interface components. Because of the hierarchical structure of the XFlash framework, we can implement web user-interfaces in a component-based manner. This approach contributes to developing reusable web user-interfaces component for different web applications.

## **2 Related work**

Schwabe [2] proposed the object-oriented hypermedia design methodology for designing the layer structure of a web application. They proposed a series of steps to implement web applications. For each phase of development, an object-oriented approach is used to model the desired result. Daniel proposed three models to describe the web application structure, namely the conceptual model, the navigational model and the abstract user-interfaces model.

The XFlash framework incorporates both the layer structure of MVC model and the model-driven approach of the OOHDM approach. We design the navigational structure in the abstract user-interfaces model rather than the navigational model, and the navigation behavior of a web application is implemented by the web user-interfaces rather than the control layer component.

There are many systems implementations of the model-driven approaches. A typical example is the MODEL-BUILD-DEPLOY platform. The primary objective of the system is to organize and visualize the structure and components of software intensive systems. The models visually represent the requirements, subsystems, logical and physical elements, the structural and the behavioral patterns. However, it is not applicable to web application development. In the XFlash framework, we propose an approach for developers to design web user-interfaces models in a component-driven manner. This contributes to web user-interface components reusability for different web applications.

## **3 Systems architecture**

The model driven framework is implemented using the Eclipse plug-in system, which can be integrated with the Eclipse Java program development platform seamlessly.

The developers need not switch between different kinds of development environment [1]. The model-driven framework provides a visible graphical editor to help the developer design the web user-interfaces model and the corresponding XML descriptor files for the web user-interface components. In addition, the model-driven framework is able to generate the SQL statements based on the navigation model, for data retrieval using the view layer components. The following figure is the architecture of the model-driven framework - XFlash.

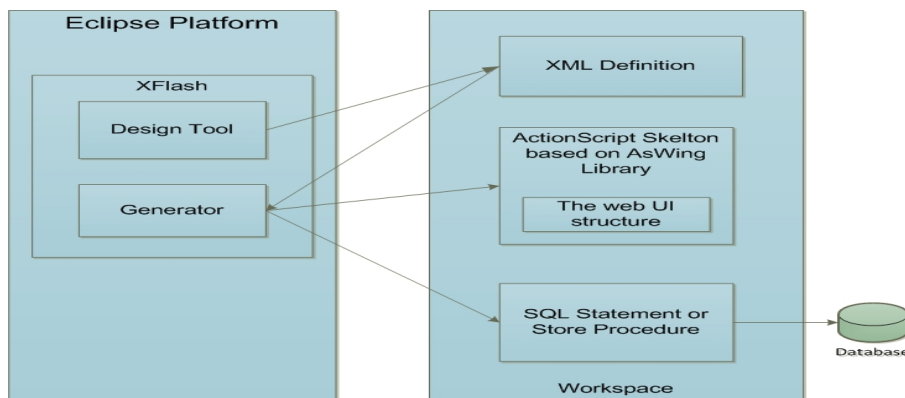


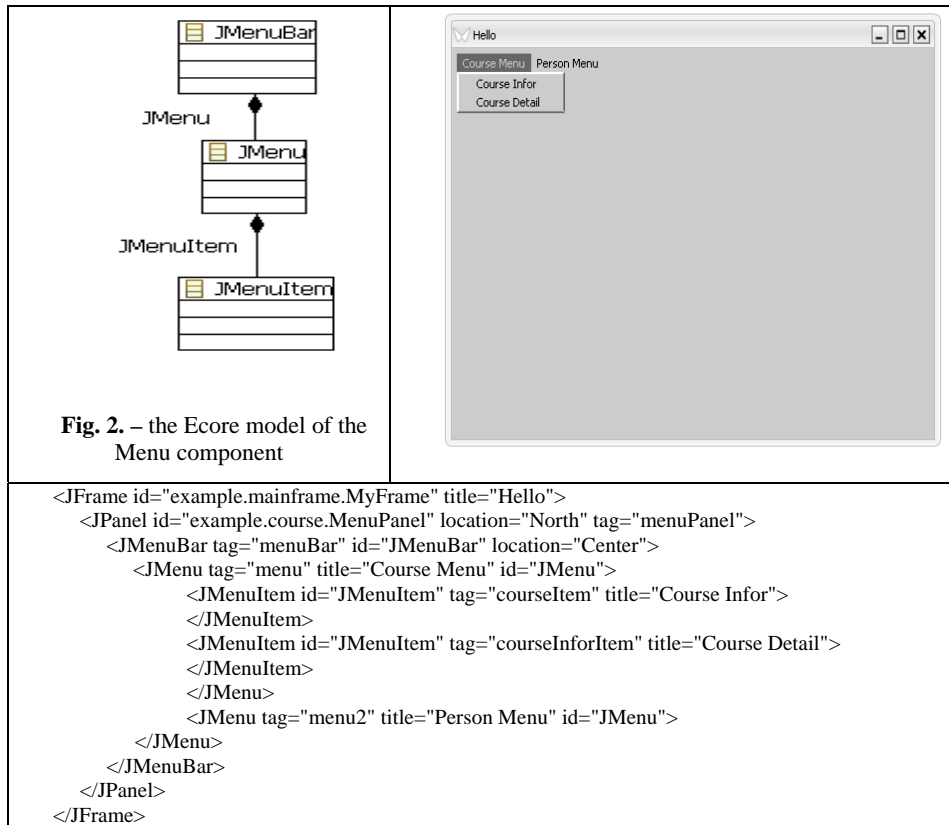
Fig. 1. – The XFlash architecture

According to the XML definition file, XFlash generates the web user-interfaces component source code automatically. Furthermore, the ActionScript file is parsed by XFlash to generate the executable web user-interfaces.

### 3.1. The XML schema of the web user-interface components

In terms of the web user-interfaces component, we define four basic component schemas: The JFrame Component, the JPanel Component, the JMenu Component and the primary component (basic components such as a button, which cannot contain other components).

To demonstrate features of basic components, we define the Menu component, which appends a menu bar to the main frame. Fig. 2 is the Ecore model of the Menu component, which is similar to the JMenu component of Java Swing. The figure below shows the source code example of Menu component, and the screen painter is the final result generated from the examples.



### 3.2. The navigational model XML description file

In OOHDM, an application can be seen as a navigational view over the conceptual model. This view is built during navigational design taking, into account the types of intended users, and the set of tasks performed by the application [3]. Usually, the navigational model describes the navigational structure and the data retrieval structure of the navigational space. However, the navigational model complicates application development in terms of the followings:

- 1) In terms of Model-View-Control (MVC) framework, the boundary between the view layer and the Control layer is unclear.
- 2) The Control layer takes charge the behavior of View layer, this breaks the major aspects of MVC framework for separating the view and control layer.

In the XFlash framework, we modify the navigational model of the OOHDM approach. The navigational model just describes the data retrieval structure. For the navigational structure, we describe the details in the abstract user interface model.

Fig. 3 is an example of navigational model structure. In this context, the relationship between professor and course is one-to-many. We present this relationship with a link class. XFlash stores this link class as an XML descriptor file.

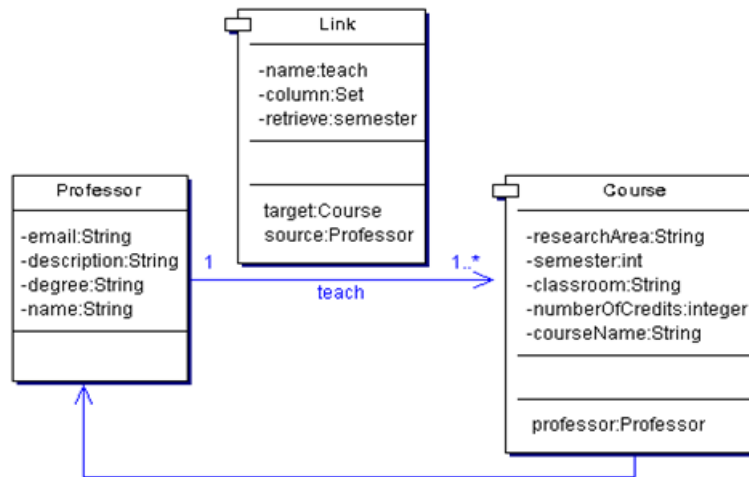


Fig. 3. – An example of navigational model

The sample XML descriptor file for the navigational model is shown in the following example (Here we assume that the database has been built according to the conceptual model). Furthermore, based on the XML file, XFlash generates the SQL statements used by the data engine<sup>1</sup> for data retrieval : select course.name, course.semester, course.classroom from course join professor where professor.name = @parameter/.

```

<Link name='teach'>
  <target name='Course'>
    <column id='name'>
    <column id='semester'>
    <column id='classroom'>
  </target>
  <source name='Professor' retrieve='name' />
</Link>

```

### 3.3. The visible graphical editor

By using the Eclipse Modeling Framework and Eclipse Graphical Framework, we develop a Visible Graphical Editor, which helps the developer to design the abstract web user-interfaces model. Using the web user-interface model, the XFlash framework generates the XML descriptor files, which are used for generating the web user-interface components.

<sup>1</sup> Here, we use Hibernate framework to implement the data engine.

For traditional web applications, a web page is the basic element of the whole web application. Here, the web page is generated as a Flash movie rather than a traditional web page. Because of the hierarchical structure of Flash movie, our web user-interface component structure is organized as a tree structure, which uses the Ecore model to represent the web-interface model.

In the Ecore model, the aggregation relationship describes the inclusion relation among the classes. In terms of the web application development, we also provide an inclusion relation among the web nodes. For example, for an academic web application, the “MainMenu” represents the entry point of the web application; it consists of “LaboratoriesMenu”, “CourseMenu”, “PersonalCategoryMenu”, and “ResearchMenu”. Fig. 4 is the aggregation relation structure of the web application. In the XFlash framework, developers can design the model using a visual graph editor. Fig. 5 shows the web interface components generated using the XFlash framework.

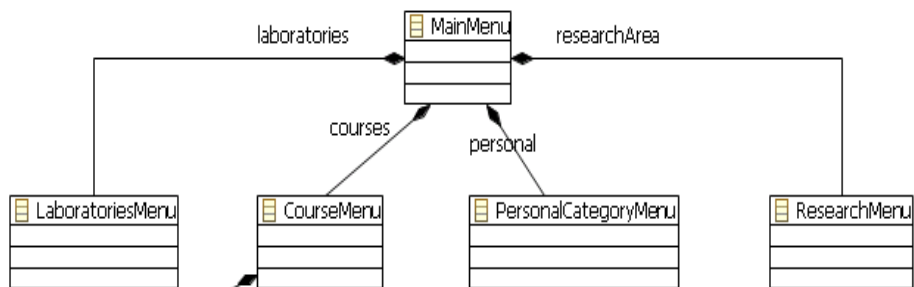


Fig. 4. - The aggregation relation among web node class

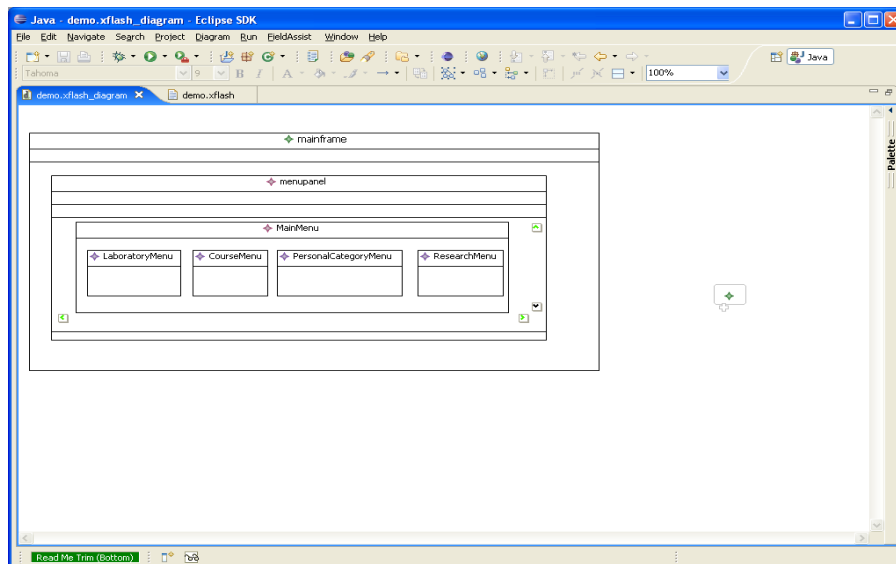


Fig. 5. - The Menu Structure Layout designed by XFlash

According to the layout structure shown in Fig. 5, XFlash generates the XML descriptor file (as shown in Fig. 6).

```

<?xml version="1.0" encoding="UTF-8"?>
<null:Diagram xmi:version="2.0" xmlns:xmi=http://www.omg.org/XMI
  xmlns:null="null">
  <JFrame id="mainframe">
    <JPanel id="menupanel">
      <JMenuBar id="MainMenu" tag="menuBar" location="CENTER">
        <JMenu id="LaboratoryMenu" title="Laboratory" tag="lMenu"/>
        <JMenu id="CourseMenu" title="Course" tag="couseMenu"/>
        <JMenu id="PersonalCategoryMenu" title="Personal"
          tag="personalMenu"/>
        <JMenu id="ResearchMenu" title="Research" tag="researchMenu"/>
      </JMenuBar>
    </JPanel>
  </JFrame>
</null:Diagram>

```

Fig. 6. - The XML descriptor file

### 3.4. Application Code Generator

The Application Code generator is the engine used to generate ActionScript source code for the Flash movie. The Eclipse JET framework is used to implement this module. The application code generator consists of three modules, which are shown in Fig. 7 :

- 1) Template module: this module defines the basic syntax of ActionScript.
- 2) The code generator : this module is the executer of code generation. The main task of the code generator is to parse the XML file, and combine the template with the data.
- 3) The compiler: the task of the compiler is to compile the ActionScript source code into a Flash movie file.

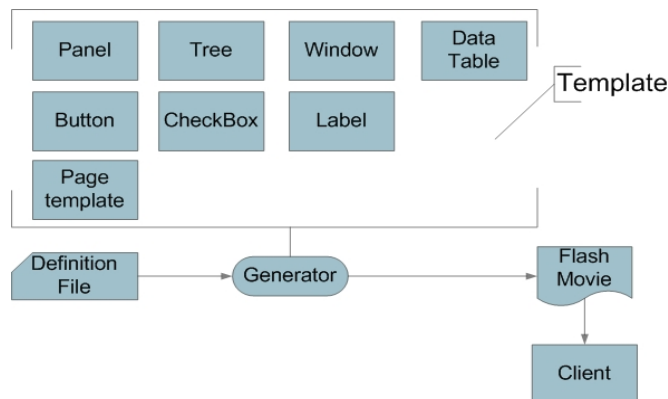


Fig. 7. - The code generator

The generator takes a single definition file, the web user-interface model, as the input file. In order to generate the output file, the generator uses a set of nested templates, and binds the template with XML data by inserting data into the hot spot of the template file.

In the XFlash framework, we design a special component called the container component. It contains the other component to be included as its child component. In order to improve the component reusability, we implement every container component model as an independent component, which not only contains other components but also contains the other container components as well.

### 3 Conclusions

In this project, we have developed a model-driven methodology for web application development. In terms of web user-interfaces development, we have implemented a generator for generating the user-interfaces components from the XFlash framework. Different from traditional web user-interface development using different computer languages, we implement the web user-interfaces using a single computer language - the ActionScript, and organize the structure of web interfaces using design patterns. Our approach contributes to the reusability of web user-interface components in different applications. Finally, our approach generates flash components to implement the web user-interface elements.

### ACKNOWLEDGMENT

This project is supported by UGC CERG grant no. PolyU 5200E/04 of the HKSAR

### References

1. Cheung, R.: A Design Automation Framework for Web-based Applications. In: Proceedings of the 3rd International Conference on Computer and Information, pp11--18, USA (2004)
2. Schwabe, D., Rossi, G.: An Object Oriented Approach to Web-based Application Design. :Theory and Practice of Object Systems., vol. 4, pp 207--225 (1998)
3. Diaz, A., Gordillo, S., Rossi, G.: Specifying Navigational Structures by Querying Hypermedia Design Models. In: Proceedings of the 3rd Basque International Workshop on Information Technology, pp125--130, France (1997)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns—Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series (1995)
5. Göbel, S.: An MDA Approach for Adaptable Components. In: First European Conference on Model-Driven Architecture – Foundations and Applications, pp74--87, Germany (2005)