
Project : XML Editor for STEP-BOOK
[Design Specification]

Class name : (CEE6583) Object-Oriented and Multimedia

Programming in Engineering

Name : Injoong Kim

Email : gtg200c@prism.gatech.edu

Date : 04/05/02

Contents

- 1. Development/Runtime Environment**
- 2. Problem Statement**
- 3. Class Design (version 2.0)**
- 4. File Handling (Imports/Exports)**
- 5. XML parser**
- 6. Reusability of class**
- 7. Assumption & Limitation**
- 8. Test Plan**

Based on the problem description in Functional Specification, the design document will explain the design rationale and implementation details.

1. Development/Runtime Environment

Object Oriented Programming

As commented in Functional Specification, I decided to use Object Oriented Programming Technology based on the following three reasons.

- (1) This system can be divided as two big blocks. One is XML part and the other is EXPRESS part. And technically both XML and EXPRESS are based on object oriented concept.
- (2) There are many reusable C++ or JAVA parsers for XML and EXPRESS.
- (3) After this class project, version-ups are inevitable. So, I want to use object orient programming technology because the OOP based systems are easier to upgrade than the structural programming based systems

Java

I decided to use Java based on the following two reasons.

- (1) The systems related to this class project are implemented with Java.
- (2) I will use XML parsers for validating of XML files. And there are many Java based XML parsers.

Java based XML parser : IBM xml parser for Java 1.1.9 (Alphaworks), Xerces Java Parser , The Oracle XML Developer's Kits (XDK) 1.4.4,

Runtime Environment

This system is designed for Window and Unix system. And It will run only on the machine in which jdk1.3.1_01 is installed.

2. Problem Statement

The purpose of this class project is to create XML editor for STEP-BOOK with which STEP-BOOK developer can easily make XML file for STEP-BOOK.

To help STEP-BOOK developer, this system should import express file, show entity information and show attribute information. And XML editor, this system should create XML files which contain EXPRESS information and GUI widget information.

3. Class Design (version 3.0)

This system can be divided as two big parts. One is XML part and the other is EXPRESS part.

3.1 XML part

As mentioned the above, XML are based on object oriented concept. So, according to LKSoft DTD I can design classes

* LKSoft DTD : acme/gtg200c/cee6583/project/LKSoft.dtd

(1) Using Vector (Vector class)

In this project I should deal with undefined number of XML elements as a group, so it will be very convenient to use collection. A collection is simply an object that groups multiple elements into a single unit. Collections are used to store, retrieve and manipulate data, and to transmit data from one method to another.

Vector is one of collections. It is a kind of array for objects but it is not necessary to define the number of objects because Vector automatically increase and decrease the number of objects.

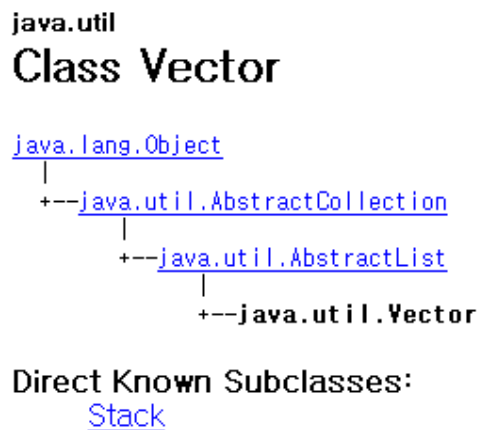


Fig. 3-1 Class Vector

3.2 Class design for XML element of GUI widgets



Fig. 3- 2 XML element of GUI widgets

Fig. 3-2 shows the XML element for GUI widgets. In this project I just want to use circled XML elements because circled elements are basic one and the other elements can be added if necessary.

The following table shows the class design. As mentioned the above, I will use Vector class to deal with the undefined number of XML information as a group.

During the design of classes, I found two interfaces and one mother class. The three classes are

like the below

(a) XmlPrint interface is introduced. All the classes representing XML Elements should implements this interface, so they show its own XML code.

(b) XmlVectorAction interface is introduced because all the classes inherited from Vector have the similar functions. It will be good for reusability and compatibility of the code.

(c) XML elements for GUI widgets have the similar information, so XmlWidget class is introduced.

Table 3-1 interfaces & mother class for GUI widgets

Class	Member Variable	Member Function
XmlString :interface		printXml() getXml()
XmlVectorAction() : interface		addCtrlElement () deleteCtrlElement () returnIndex() printXml() getXml() printObjectList()/getObjectList() clearAllObject()
XmlWidget	name page_name tap_name group_name display_name entity_name parent_name	setName ()/getName () setPage_name ()/getPage_name () setTab_name ()/getTab_name() setGroup_name()/getGroup_name() setDisplay_name()/getDisplay_name() setEntity_name ()/getEntity_name () setParent_name ()/getParent_name () printInfo() / getInfo()

Table 3-2 Class design for XML element of GUI widgets

Class	Member Variable	Member Function
XmlEditBox (inherited from XmlWidget) (implement XmlPrint)	attribute_name	setAttribute_name() getAttribute_name()

XmlVectorEditBox (inherited from Vector) (implement XmlVectorAction)		
XmlLinkedField (inherited from XmlWidget) (implement XmlPrint)	attribute_name display_attribute_name1 display_attribute_name2	setAttribute_name() getAttribute_name() setDisplay_attribute_name1() getDisplay_attribute_name1() setDisplay_attribute_name2() getDisplay_attribute_name2()
XmlVectorLinkedField (inherited from Vector) (implement XmlVectorAction)		
XmlAggregateView (inherited from XmlWidget) (implement XmlPrint)	attribute_name display_attribute_name1 display_attribute_name2	setAttribute_name() getAttribute_name() setDisplay_attribute_name1() getDisplay_attribute_name1() setDisplay_attribute_name2() getDisplay_attribute_name2()
XmlVectorAggregateView (inherited from Vector) (implement XmlVectorAction)		
XmlCustom (inherited from XmlWidget) (implement XmlPrint)	java_class_name	setJava_class_name () getJava_class_name ()
XmlVectorCustom (inherited from Vector)		

Table 3-3 Class design for XML element of Group of GUI widgets

Class	Member Variable	Member Function
XmlGroup (inherited from Vector) (implement XmlPrint)	name page_name tab_name group_name show	setName() / getName() addEditBox() / deleteEditBox() addLinkedField() / deleteLinkedField() addAggregateView() deleteAggregateView() addCustom() / deleteCustom()
XmlVectorGroup		

(inherited from Vector)		
(implement XmlVectorAction)		

3.3 Class design for XML element of Views

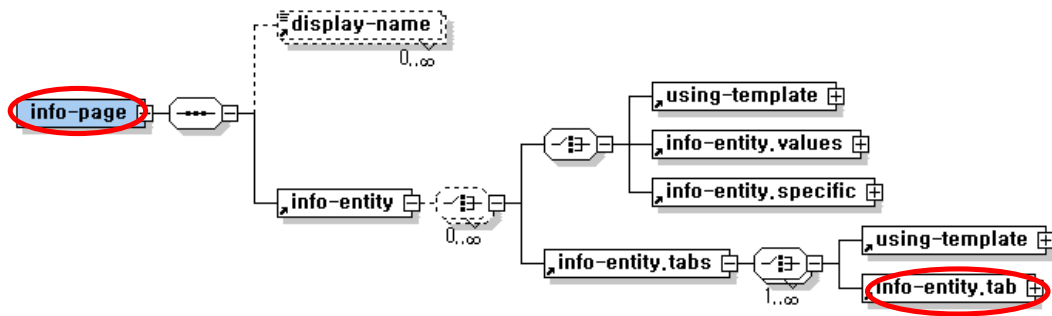


Fig. 3- 3 XML element of Views

Fig. 3-2 shows the XML element for Views. I will create the classes of circled XML elements because they are enough to carry out my lab project.

Table 3-4 Class design for XML element of Views

Class	Member Variable	Member Function
XmlTab (implement XmlPrint)	page_name tab_name entity_name attribute_name sub_entity_name1 sub_attribute_name 1 sub_entity_name2 sub_attribute2	setPage_name () / getPage_name () setTab_name () / getTab_name () setEntity_name () / getEntity_name () setAttribute_name () getAttribute_name () setSub_entity_name1 () / getSub_entity_name1 () setSub_attribute_name1 () / getSub_attribute_name1 () setSub_entity_name2 () / getSub_entity_name2 () setSub_attribute_name2 () / getSub_attribute_name2 () addGroup() / deleteGroup()
XmlVectorTab (inherited from Vector)		

(implement XmlVectorAction)		
XmlInfoPage (implement XmlPrint)	page_name entity_name display_name	setPage_name () / getPage_name () setEntity_name () / getEntity_name () setDisplay_name()/getDisplay_name()
XmlVectorInfoPage (inherited from Vector) (implement XmlVectorAction)		addGroup() / deleteGroup() addTab() / deleteTab()
XmlVectorGroupInfPage		

3.4 EXPRESS part

List of nouns

- (1) EXPRESS_schema
- (2) EXPRESS_entity
- (3) EXPRESS_attribute

Association between classes

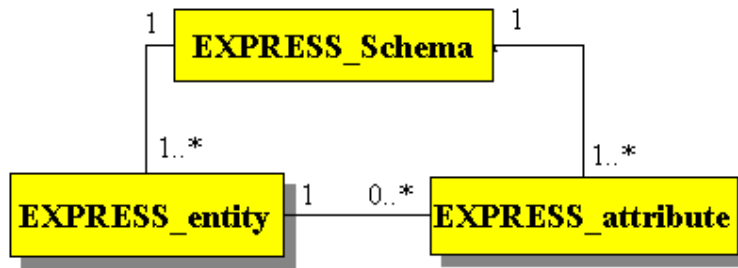


Fig 3-4 Class diagram for EXPRESS information

In one EXPRESS schema, there are many entities and attributes. And one entity has more than zero attributes.

Table 3-5 Class design for EXPRESS information

Class	Member Variable	Member Function
ExpSchema	schema_name group_entity group_attribute	setSchema_name() getSchema_name() show_all_inherited_attribute_list_of_entity ()
ExpEntity	entity_name	setEntity_name()

	super_entity_name	getEntity_name() setSuper_entity_name() getSuper_entity_name()
ExpVectorEntity (inherited from Vector)		add_entity() show_entity_list() show_entity_inheritance()
ExpAttribute	attribute_name entity_name type number_type	setAttribute_name() getAttribute_name() setEntity_name() getEntity_name() setType() getType() setNumber_type() getNumber_type()
ExpVectorAttribute		add_attribute() show_attribute_list_of_entity()

4. File Handling (Imports/Exports)

This system need pseudo EXPRESS file as an input and create XML file as an output.
So, file handling is very important in this system.

To handle files in Java, understanding the following four classes is necessary.

@<http://java.sun.com/docs/books/tutorial/essential/io/filestreams.html>

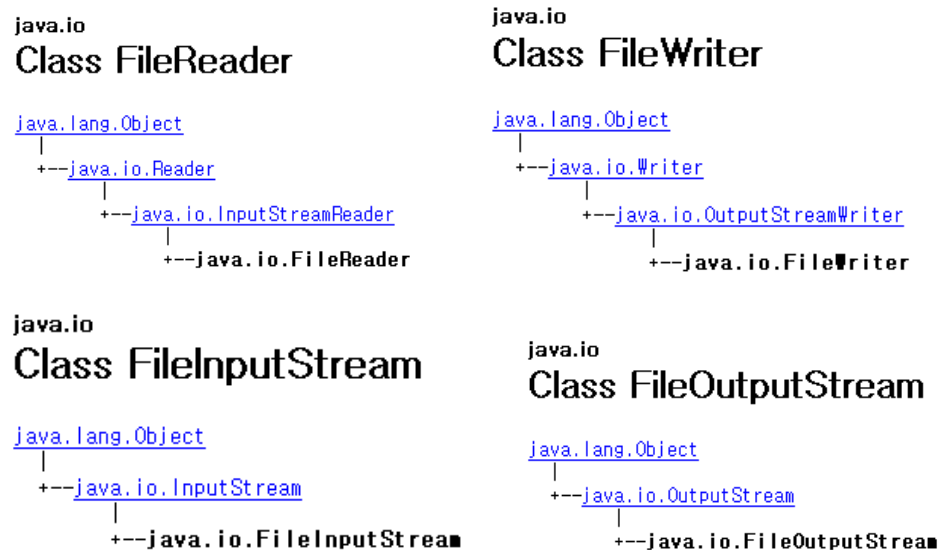


Fig. 4-1 Java Classes for file handling

The following Copy program uses `FileReader` and `FileWriter` to copy the contents of a file named `farrago.txt` into a file called `outagain.txt`:

```
import java.io.*;

public class Copy {
    public static void main(String[] args) throws IOException {
        File inputFile = new File("farrago.txt");
        File outputFile = new File("outagain.txt");

        FileReader in = new FileReader(inputFile);
        FileWriter out = new FileWriter(outputFile);
```

```
int c;

while ((c = in.read()) != -1)
    out.write(c);

in.close();
out.close();
}
}
```

The key point in file handling is to distinguish between stream and stream reader/writer.

5. Parser

5.1 XML Parser

To validate create XML file I will use IBM xml parser for Java 1.1.9. In this class project, I will use XML parser just for validation. But after this class project I will use other functions of XML parser. For example, with XML parser I could create XML FILE importing module, then I could edit already existing XML document.

** Site for IBM xml parser for Java: <http://www.alphaworks.ibm.com/tech/xml4j>*

[Description about IBM xml parser for Java]

XML Parser for Java is a validating XML parser and processor written in 100% pure Java; it is a library for parsing and generating XML documents. This parser easily enables an application to read and write XML data. Version 3.2.0 incorporates support for the following:

- W3C XML Schema Recommendation 1.0
- SAX 1.0 and 2.0
- DOM Level 1, DOM Level 2, and some features of DOM Level 3 Core Working Draft
- JAXP 1.1

[Functions of IBM xml parser for Java]

- Validate XML document
-

-
- Reading XML document
 - Writing XML document
 - Creating DOM tree : searching XML information
 - Modifying XML information
 - Adding XML information
 - Deleting XML information

5.2 EXPRESS Parser

This system just supports importing pseudo EXPRESS files instead of real EXPRESS files because it is very hard to interpret EXPRESS files without EXPRESS parser. Next version can support importing real EXPRESS files with the parser.

**Site of Open Source EXPRESS Parser : <http://osexpress.sourceforge.net/>*

[Description about Open Source EXPRESS Parser]

- Uses ANTLR language recognition tools
- Provides code for parsing EXPRESS and EXPRESS version 2
- Generates an AST (Abstract Syntax Tree)
- Includes template files for building applications
- Includes pretty-printer application for generating indented EXPRESS text from an AST
- Includes application for generating an XML Schema from an EXPRESS schema

6. Reusability of class

For the purpose of clarity and modularity, I separated classes to two groups. One is Xml group and the other is Express group. To do this I will use package technique of Java.

(1) Xml package : the group of classes related to XML

(2) Express package : the group of classes related to EXPRESS

And as mentioned at the above, two interfaces and one mother class are introduced for the reusability and compatibility of the code.

7. Assumption & Limitation

This system does not support all the expressions of EXPRESS, like inverse and selection type and also does not support all the elements of LKSoft DTD, like check-box and combo-box. But it support enough expressions and elements to carry out my lab project and to overcome this limitation it is developed based on extensible class platform.

In this project I did not use XML parser and EXPRESS parser, so this system just read pseudo EXPRESS files instead of real EXPRESS files and can not read XML files. I think that using parsers is the only way to overcome this problems.

8. Test Plan

Test will be carried out only on the machines whose OS is Window2000 or Unix.

I have two test cases. One is STEP-BOOK of spring-systems and the other is STEP-BOOK of PWA. In other words, I have EXPRESS files and XML files of spring-systems case and PWA case.

Test of valid XML files

After making XML files with XML Editor for STEP-BOOK, XML files easily can be checked by Microsoft Internet explorer.

Test of STEP-BOOK Operation

The purpose of the system is creating XML files for STEP-BOOK. So, we can test the system seeing the STEP-BOOK. The procedure of making STEP-BOOK is the like the Figure 5.1

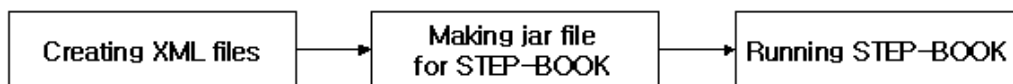


Figure 5.1 XML files for STEP-BOOK

The following two figures show STEP-BOOKs of spring-systems and PWA.
