

AXE Component Proposal

A WST Component

This document is a proposal for a new WST component codenamed AXE.

1 Introduction

Since the XML specification, we can find powerful XML syntax editors. Those editors check your XML file, according to the core syntax, sometime a DTD or a schema. Unfortunately, XML editing based on forms, and graphical widgets keeps an heavy development task.

This proposal describes the Anyware Technologies XML Editing plateforme (named AXE), which supplies interesting features to produce easily and quickly, powerful eclipse XML editors, based on graphical pages. Those editors look like the plugin.xml editor (flat UI style).

2 Overview

2.1 Concept.

An AXE editor is composed of several pages, containing widgets.

A widget is a SWT element bound, using XPath annotation, to any value in the XML file.

The AXE platform is an eclipse plugin, providing extension point to declare easily the pages, and a basic XML widget library to edit any part of the XML file based on SWT or Forms (from the org.eclipse.ui.forms plugin).

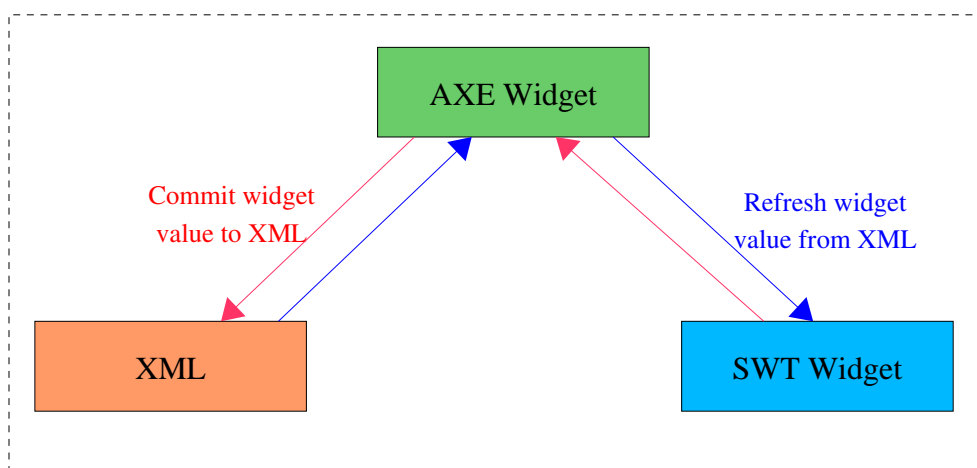


Figure : 1. AXE widget behavior

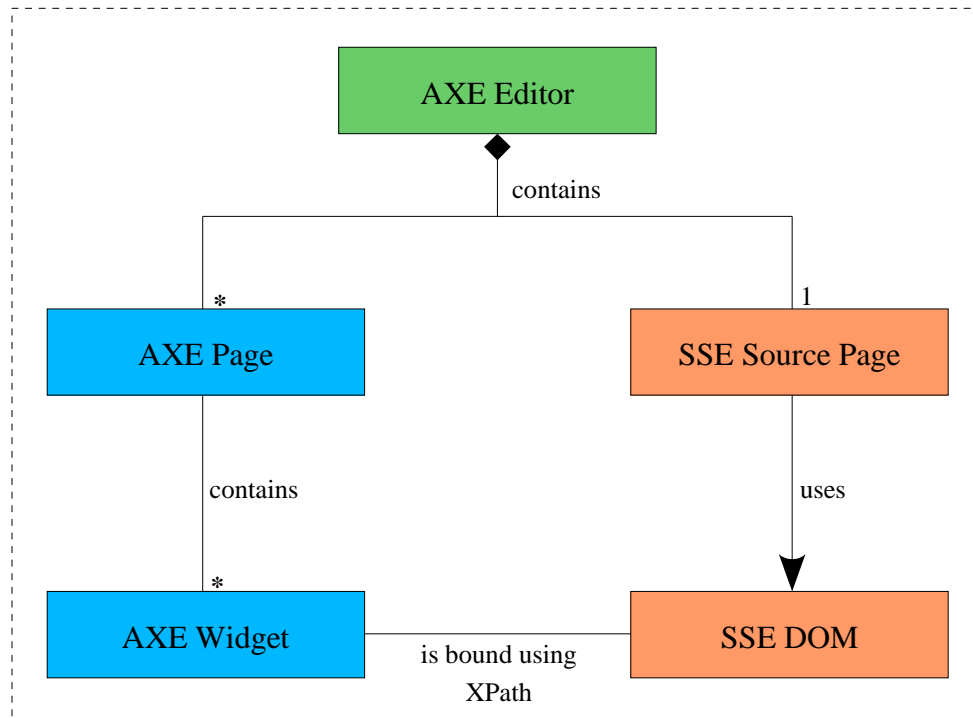


Figure : 2. An AXE Editor

2.2 AXE Features

The AXE platform is designed to gather the maximum of common features, that are needed for a powerful XML edition component. These features are listed below :

- Extension point to define your edition pages
- Powerful SWT/XML widgets to display data : Tree, list, class chooser, ...
- Mechanism to update data in all pages.
- Easy way to fill widgets with data contained inside file.
- A real integrated eclipse editor (undo, file association...).
- An extensible mechanism to add you own behavior for any widget in any page.
- Taglib extension capability.
- Navigation across pages.
- Syntactical control of values : integer, ...
- I18N support.

2.3 AXE usage.

To create an AXE Editor, depends on the AXE platform, then describe your pages using SWT or Jelly Scripts.

Jelly :

Jelly is a tool for turning XML into executable code. So Jelly is a Java and XML based scripting and processing engine.

JellySWT :

JellySWT is a simple Jelly library which can be used to create SWT user interfaces. It allows XML documents (Jelly scripts) to be used to define the layout and rendering of SWT front ends which avoids lots of mundane Java coding, using markup to define the view of your front end and allowing you to bind to Java code for the business objects and models.

This mechanism uses separation of concerns and MVC ideas from web applications, allowing the rendering of your SWT front end to be easily transformed (since its XML) into different styles while leaving your model and business objects untouched. It also allows different views to be constructed independently of your models.

UI created with Jelly are more evolutive, more readable and logic is separated from the display.

AXE uses the JellySWT tag library (taglib) but also defines a taglib for the Forms API and the AXE widgets.

The Jelly plugin defines classes to manage Jelly scripts and an extension point to add new Jelly taglibs.

3 Architecture

The following schema summarizes the AXE architecture :

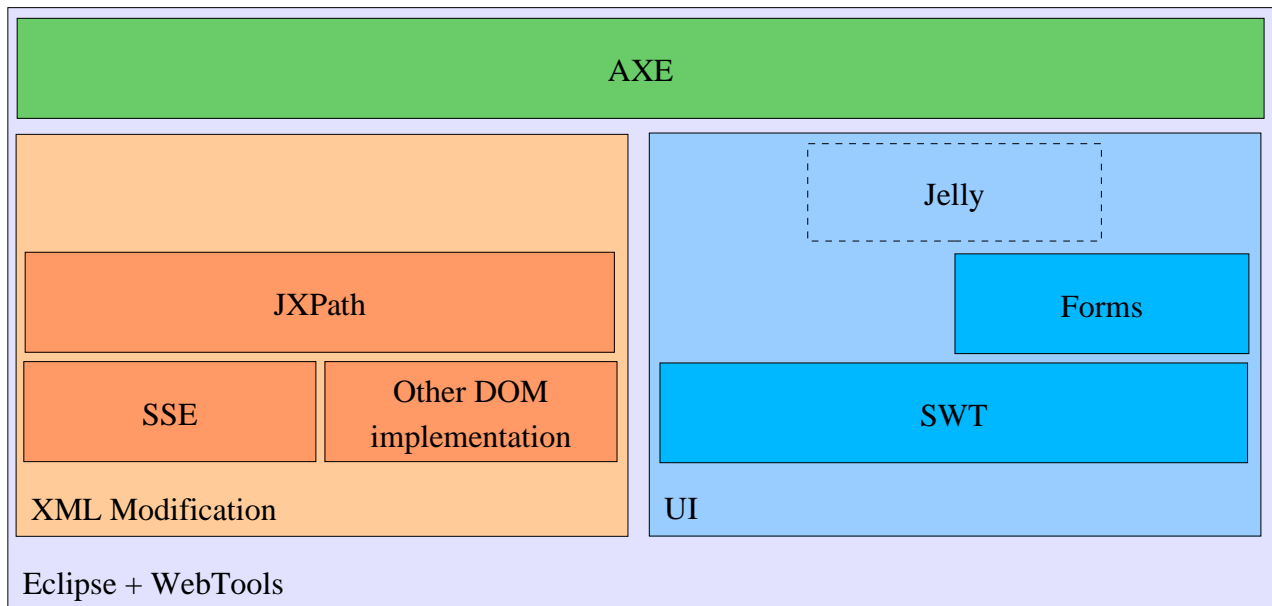


Figure : 3. AXE Architecture

4 Plugin structure

4.1 Contents

→ **org.apache.commons**

The Apache Commons libraries plugin.

→ **org.eclipse.wst.jelly**

This plugin brings Jelly helpers and extension points to add new Jelly taglibs.

Extension points :

- org.eclipse.wst.jelly.taglibs : used to add new Jelly taglibs

→ **org.eclipse.wst.axe**

The AXE core and the basic widgets.

Extension points :

- org.eclipse.wst.axe.validators : used to add input validators

- `org.eclipse.wst.axe.configuration` : used to add pages to an AXE editor and customize its behavior.

→ `org.eclipse.wst.axe.java`

The widgets linked with the JDT : Class selection widget, Method selection widget...

Those widgets are not included in the core plugin for editors that are not using JDT (for RCP applications for example).

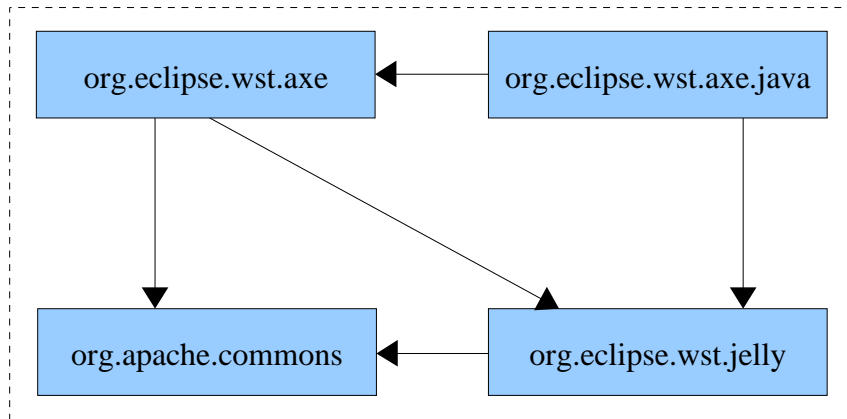


Figure : 4. AXE widget behavior

4.2 Dependencies

Apache Commons :

- Jelly : dependency from AXE
- Jexl : dependency from Jelly
- XPath : dependency from AXE
- BeanUtils : dependency from Jelly and XPath
- Collections : dependency from Jelly and XPath
- Logging : dependency from Jelly and XPath

Dom4J : dependency from Jelly

5 Initial Contribution

Anyware Technologies proposes to contribute to the web tools project, by giving the AXE platform to the eclipse foundation.

6 Organization

6.1 Initial Committers

David Sciamma david.sciamma@anyware-tech.com

Olivier Prouvost olivier.prouvost@anyware-tech.com

Thomas Friol thomas.friol@anyware-tech.com

6.2 Contact

Anyware Technologies

Prologue 2

Rue Ampère BP 87216

31 312 LABEGE Cedex

FRANCE

Tél : +33 (0)5 61 00 52 90

Fax : +33 (0)5 61 00 51 46