

Eclipse Summit Europe 2006

Position Paper

Author: Martin Roberts – BT plc

Title: RCP and Eclipse prove to be a key part of the Strategy for the creation of Standards. (Also how a newbie found RCP)

Over the last few years the number of standards in all areas of computing has increased dramatically. One of the needs of those developing standards is to prove that they work. For a number of standards bodies the need to produce working solutions is part of the standards adoption process.

For the last 3 years an OASIS Technical Committee – Content Assembly Mechanism –CAM - have been working on a standard that recognises the limitations of schema based XML definition as a method of describing XML content especially in E-Commerce solutions. To help this standard two Eclipse applications have been developed by fairly non-java literate developers. One is a pure Java CAM XML Validator. The other is an Editor for the CAM Templates. These templates are reasonably complex XML documents that utilise XPath 1.0. The former application is delivered as a simple JAR file. The second is an Eclipse Rich Platform Application.

This paper will discuss both the needs of the standards community to have an Open Source platform for the delivery of early proof ‘prototypes’ and the issues around getting to grips with the whole Eclipse Platform.

Eclipse has proved to be a very useful and suitable development environment for the production of the JCAM processor and editors. The former simply utilised the fully integrated development environment to deliver a simple Java application. The ability to use both ANT, JUnit and Subversion transformed the developers expectations of a development environment from something that was painful and had a very steep learning curve into something that is intuitive and flexible enough to for the developer to be confident in his own code for the first time in 20 years.

For the Editor the choice of Eclipse was natural. After investigating other application development environments such as Smalltalk and Visual Studio, the facilities of the RCP proved to be a winner. However, moving from simple Java with a number of simple patterns to using Eclipse UI facilities was not without its problems.

Eclipse from the outside does have a huge vocabulary. Each new item seems to carry a set of concepts that must appear to some to be straight forward. However, without the large volume of examples the ability of a newbie to develop applications in RCP would be greatly restricted. Also unlike the simple java application with its ability to use simple JUnit test support. The developer for the CAM editor struggled to provide any form of test environment. He also although providing a working application has that nagging doubt that there are simpler ways of doing things if only he could get his head round the vast number of alternative ways of doing things.

RCP provides a serious alternative to other application building environments. However, it is amazingly complex. It is subtle in its facilities. There can be a sense that the developer is not quite in control of their own destiny.

How can this be addressed? Eclipse to its credit continuously provides new versions, with improvements. However, most of the documentation about the platform is in books and articles that are out of date almost at the point of writing. The introduction of Eclipse 3.2 for example led to the removal of some of the Example application templates. This reduces confidence that as a developer you are doing things correctly. Using items that are deprecated can prove to be the only way forward as often there is no indication of what alternative to use.

One major issue that RCP needs to address is the method of packaging extensions. The IDE provides some useful tools to enable dependencies to be managed. However, it is not clear from the names exactly whether the features you have used require the plug-in. This came about when the first version of the CAM Editor was created and two menu items appeared – Search and Run – neither of which appeared in the code. These appear due to some automatic contribution mechanisms that is almost unfathomable to the newbie. Tracking down the names of the plug-ins that provided these features, proved to be a very lengthy process.

The other issues that crop up as you approach an application in RCP is that there seems to be so many ways of achieving similar ends. For example, do you use ActionFactory or simple Actions for menu options? If you use something like the default exit function how do you hook in a ‘save changes’ option. The newbie struggles to walk around the maze and this can give a feeling of I shall never be able to create that professional solution I would like to create.

Finally, it is worth noting that the RCP and Eclipse have enabled two applications to be created that have enabled OASIS CAM Technical Committee to prove their ideas and is a key part of the strategy to create a useful standard.