

Openness in Systems Engineering with Eclipse

Dr. Michael Jastram (CEO at Formal Mind GmbH)
michael.jastram@formalmind.com

Presentation Submission for ProSTEP iViP Symposium

Eclipse¹ is an open source framework for building platform-independent GUI applications. It is managed by the Eclipse Foundation (a non-profit organization), which ensures that official Eclipse projects are interoperable and follow certain intellectual property guidelines.

Eclipse was founded in 2002 by IBM as a strategic decision. By now, it consists of over 70 projects, consisting of over 40 million lines of code, with over 50 companies actively participating in its development, and countless organizations using it in a production environment.

Eclipse was initially designed as an Integrated Development Environment for Java and later for Software in general. But it recently gained attention in the embedded market. For instance, Airbus decided to co-sponsor the development of Topcased², a software environment primarily dedicated to the realization of critical embedded systems consisting of hardware and software.

Open Source in general allows organizations to remedy the risk of being dependent on one single vendor. This includes the risk of the feature set provided: users can add missing features themselves or commission their inclusion to any competent party, rather than having to rely on the vendor to implement it. It further includes the risk of maintenance and long-term support.

Eclipse in particular provides a solid, mature and open platform for desktop applications with a rich ecosystem. Many Eclipse offerings are ready to be used “as is”, thereby offering great

cost savings. An example of this is the Java development environment, which is of production quality. Eclipse also makes it easy to build custom tools. The Eclipse Modeling Framework, for instance, generates code from data models, thereby allowing the creation use user interfaces with relatively little effort. Eclipse is an integration platform, making it easy to combine various “plug-ins” into a seamless package. For instance, there are various plug-ins for version control. These can be combined with any file-based application and provide an integrated user interface, even though the plug-ins were not inherently designed to interoperate. And last, the Eclipse Public License is business-friendly and allows the commercial reuse and the combination of commercial and open source elements.

A number of Eclipse-based environments for systems engineering exist. We already mentioned Topcased, others include Unicaise or the Open System Engineering Environment (OSEE). Acknowledging the need for domain-specific tools, the Eclipse automotive working group started to develop an industry-specific tool.

In this talk, we demonstrate how Eclipse can be used as an integration platform for systems engineering. We will introduce a few projects from the Eclipse ecosystem like UML (Papyrus), Modeling (EMF) or requirements (RMF³). We focus on RMF (Requirements Modeling Framework) as a case study on how the Eclipse ecosystem can be leveraged in a business environment. RMF is a clean-room implementation of the open ReqIF standard, which is currently

1 <http://eclipse.org>

2 <http://www.topcased.org/>

3 <http://eclipse.org/rmf>

being adopted by various tool vendors: The currently ongoing ReqIF Implementor Forum⁴, which is organized by ProSTEP iViP, will ensure that the various ReqIF implementations will properly function together. We will look at both the technical and business implications.

But the real power of Eclipse stems from the fact that custom environments can be created easily thanks to its plug-in mechanism. This also allows to incrementally introduce Eclipse in an organization. For instance, requirements management could be introduced through RMF. Interoperability with existing systems (e.g. IBM Rational DOORS) is possible thanks to the ReqIF standard. Step by step, the work environment can be enriched with additional plug-ins, both standard and custom-build. For instance, a standard modeling environment could be added, and a custom plug-in could provide traceability between requirements and their corresponding model elements. Step by step, more functionality can be added.

From a business point of view, this approach promises cost savings and prevents vendor lock-in. To understand the value, we will look at the openETCS project⁵, which is an ITEA2 EU-funded project. The purpose of this project is the development of an integrated modeling, development, validation and testing framework for leveraging the cost-efficient and reliable implementation of the European Train Control System (ETCS), based on open source technologies. While the technology choice has not yet been finalized, Eclipse is a strong candidate for realizing this project, and it being open source is a core requirement. We will present the implications of such an open platform from a business point of view for the parties involved, which are customers (e.g. Deutsche Bahn), equipment manufacturers (e.g. Siemens) service providers (e.g. Formal Mind) and, of course, the EU and its citizens.

4 <http://www.prostep.org/de/projekte/reqif-implementor-forum-req-if.html>

5 <http://www.itea2.org/project/index/view/?project=10135>