



Chapter II

Adopting Open Source Development Tools in a Commercial Production Environment: Are We Locked in?

Anna Persson, University of Skövde, Sweden

Henrik Gustavsson, University of Skövde, Sweden

Brian Lings, University of Skövde, Sweden

Björn Lundell, University of Skövde, Sweden

Anders Mattsson, Combitech AB, Sweden

Ulf Ärlig, Combitech AB, Sweden

ABSTRACT

Many companies are using model-based techniques to offer a competitive advantage in an increasingly globalised systems development industry. Central to model-based development is the concept of models as the basis from which systems are generated, tested, and maintained. The availability of high-quality tools and the ability to adopt and adapt them to the company practice are important qualities. Model interchange between tools becomes a major issue. Without it, there is significantly reduced flexibility and a danger of tool lock-in. We explore the use of a standardised interchange format (XMI) for increasing flexibility in a company environment. We report on a case study in which a systems development company has explored the possibility of

complementing its current proprietary tools with open-source products for supporting its model-based development activities. We found that problems still exist with interchange and that the technology needs to mature before industrial-strength model interchange becomes a reality.

INTRODUCTION

The nature of the information systems development industry is changing under the pressures brought about by increased globalisation. There is competition to offer cheaper but higher quality products faster. To stay competitive, many companies are using model-based techniques to offer rapid prototyping, fast response to requirements change, and improved systems quality. Central to model-based development is the concept of models as the major investment artefacts; these are then used as the basis for automatic system generation and test. Tools for the development, maintenance, and transformation of models are therefore at the heart of the tool infrastructure for environments which support model-based development practice.

One potential danger for companies is tool lock-in. Tool lock-in exists if the models developed within a tool are accessible only through that tool. It has long been recognised that the investment inherent in design artefacts must be protected against tool lock-in, not least for maintenance of a long-lived application. Such lock-in effects are recognised as a risk, which can have severe consequences for an individual company (Statskontoret, 2003). The tool market is dynamic, and there is no guarantee that a tool or tool version used to develop a product will remain usable for the lifetime of the product (Lundell & Lings, 2004a, 2004b). In order to protect against such problems, models must be stored together with the version of the tool with which they were created. Even this is not guaranteed to succeed — hardware changes may mean that old versions of tools can no longer be run — unless hardware is also maintained with the tool. Such lock-ins are therefore undesirable for tool users. This may not be the case for some tool vendors, who may view lock-in as a tactic to ensure future business by keeping customers tied to their products (Statskontoret, 2003).

The availability of high-quality modelling tools and the ability to adopt and adapt them to a company context are also important qualities. A variety of different development tools can be applied during a systems development project, including tools for the design of UML diagrams, tools for storing models for persistence, and tools for code generation (Boger, Jeckle, Mueller, & Fransson, 2003). The ability to seamlessly use and combine the various tools used within a project is highly desirable (Boger et al., 2003). The reality for many designers is an environment in which a mix of tools is used, and many companies are considering a mix of proprietary and open source tools to flexibly cover their needs. The interchange of design artefacts between tools becomes critical in such environments. One special case of this is geographically distributed development where partners in different locations are working in different environments, using different tool sets.

Model interchange functionality can therefore significantly increase flexibility and reduce exposure to lock-in effects. There are two accepted ways in which model interchange can be undertaken: via software bridges, and via an open interchange

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/adoptng-open-source-development-tools/4384

Related Content

Replication Methods and Their Properties

Lars Frank (2005). *Encyclopedia of Database Technologies and Applications* (pp. 555-561).

www.irma-international.org/chapter/replication-methods-their-properties/11204

From 'Flow' to 'Database': A Comparative Study of the Uses of Traditional and Internet Television in Estonia

Ravio Suni (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 1477-1489).

www.irma-international.org/chapter/flow-database-comparative-study-uses/7987

A Content-Based Approach to Medical Image Database Retrieval

Chia-Hung Wei, Chang-Tsun Liand Roland Wilson (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 1062-1083).

www.irma-international.org/chapter/content-based-approach-medical-image/7959

Methodology Evaluation Framework for Component-Based System Development

Ajantha Dahanayake, Henk Soland Zoran Stojanovic (2003). *Journal of Database Management* (pp. 1-26).

www.irma-international.org/article/methodology-evaluation-framework-component-based/3288

Situational Method Engineering to Support Process-Oriented Information Logistics: Identification of Development Situations

Tobias Bucherand Barbara Dinter (2012). *Journal of Database Management* (pp. 31-48).

www.irma-international.org/article/situational-method-engineering-support-process/62031