Report from the EWMDA-2 Working Group on "MDA and Reuse"

Julian Johnson (BAE Systems, UK), **Kevin Dockerill** (BAE Systems, UK), **Joao Paulo Almeida** (Univ. of Twente, The Netherlands), **Peter Linington** (Univ. of Kent, UK), **Zhen Ru Dai** (Fraunhofer Fokus, Germany), **Oliver Sims** (Sims Architectures, UK), **Salim Bouzitouna** (LIP6, France), **Ian de Beer** (UCSSM, South Africa), **Nelly Becomo** (Lancaster Univ., UK), **Antonio Vallecillo** (Univ. of Málaga, Spain)

The group met from 13:30 to 14:15 to discuss the topic of MDA and re-use, originally to cover two main issues: (1) How to reuse in MDA, and (2) Does MDA facilitate reuse? It was agreed to concentrate on the first one, in order to be more focused and make the best use of the time allocated for discussions.

Before we went into the discussions, we felt it was useful to review the participants' definition of the terms "MDA" and "re-use". Thus, it was agreed that **MDA** includes (at least) the following concepts: Models and Transformations; Platform independence / PIM; separation of concerns: Business / Technology; Moving Intellectual Property (IP) from code to models; Design Explicitness / visibility. With regard to **re-use**, it was agreed that it might imply re-exploitation of IP across two or more applications/programmes, in a maintained way. Furthermore, re-use not only implies reuse of design/models, but also reuse of knowledge, skills etc. Finally, it was noted that re-use does not mean using many times in the same context, but using in different contexts (quote due to Clemens Szyperski?).

Once the basic terms were discussed and their scope and meaning was generally agreed, we moved onto the main points that the Workshop organizers suggested for discussion.

The first approach was to identify **re-use contexts**. The following list of potential re-use contexts was produced:
- Product line oriented (families of application types), component-based application development
- Same application PIM used for multiple target platforms (via PSM's)
- Same platform / architecture model used for different applications
- Reuse of IP in capturing an organisation's test approach as a transformation (then used to generate test info / test cases / configurations, etc.)
- Aspects / cross-cutting concerns: security, audit trails, … MDA transformations may support the development for certain aspects
- Exploitation of legacy code / applications, via modelling of wrapping

It was agreed that the **solution** might be based on applying all principles and technologies associated with MDA, with particular emphasis on
- Separation of concerns
- PIM / PSM's
- Reusable components/transformations
- Model / component repositories

This solution has clear **advantages**, such as: (a) Most cost effective development and maintenance of existing and new applications; (b) improved knowledge and IP re-use; and (c) Explicit capture of IP. However, it also presents some **disadvantages**, such as potential performance issues; the challenges of designing for reuse; cost; and it also may have organisational implications.

After that, the group also identify a set of **issues** related to re-use such as the following:
- In order to re-use an artefact we need to be able to perform a set of tasks, such as: find it; get to it to use; maintain it; manage its lifecycle; and properly handle the possible organisational changes it might imply
- Designing for re-use may incur extra costs (development, maintenance, consequential performance, etc.). Moreover, there are difficulties in predicting use contexts
- It was also agreed that re-use implies using without changes, otherwise it is not re-use. However, some products and artefacts allow customisation and extensibility in order to facilitate re-use
- There is also the difficulty of integrating legacy systems and applications into the MDA chain

Finally, some **further work** needs to be done. In particular, the following tasks were identified:
- What are the [abstraction] criteria for development of PIM's and PSM's, an abstract platform, etc?
- There is a need for MOF repository support that is integrated with other tools
- MDA support for aspect development