

Some Questions & Answers on Model Driven Development

The purpose of this week's seminar (and associated lecture) sessions is for you to familiarise yourself with MDA, and the main issues in this field.

As usual the web-site has some presentations (and papers) which provide a particular perspective (largely my own biased view on the topic).

However, as a departure from being lectured at, this final topic based week requires you to undertake research yourself, ideally in pairs (or small groups) to find out, and report, back about one of the MDA areas given below. (I will attempt to allocate these should that be necessary).

The only constraint is that, as a group, we should cover all of these, (though it may be that some can be subsumed into a single presentation) and can give a brief description of each in the evening lecture session. (The presentations can be relatively informal, and do not need to be slides, though a further condition is that each group then makes their notes available to the others).

The suggested topic areas are:

- What are the principle concepts of model driven development?
- How can MDA or MDD be used to develop software?
- What are the advantages (strengths) of model driven development?
 - Link from Paul:
 - <http://www.theenterprisearchitect.eu/archive/2009/11/25/15-reasons-why-you-should-start-using-model-driven-development>
- What are the disadvantages (weaknesses) or issues that have been suggested for model driven development?
 - (Thanks to Sam Mullins for supplying these)
 - Generated Code can be inflexible, especially in finer details and GUIs.
 - Limited by MD engineering tool.
 - Flexibility is handled in DSLs and these are intrinsically abstract with lots of hard coded things.
 - Not designed for pure programmers or pure business engineers but a mix of the two. These are hard to find.
 - Version control and models is less than perfect.
 - Lots of trust on the tool.
 - Have to design with limitations in mind.
 - The factories are domain specific, crossing more than one domain can be hard.
- How could the model driven development approach be improved upon?
 - We specifically discussed the CIM phase (see presentations)
- What would you need to consider if you wanted to adopt a model driven development approach?
 - Site suggested by Ross:
 - <http://www.agilemodeling.com/essays/readyForMDA.htm>
- What support is there for model driven development, and does it apply equally to all aspects of the software development life-cycle?
 - Thanks to Matthew for suggesting these and for the link
 - The standard for model development in MDA is to use UML, so any tools that are capable of UML should support MDA, with all the UML diagrams or at least a sub-set.
 - An MDA tool should be able to support the complete life-cycle.

- However an MDA tool can be one or more of the following types:
 - Creation tool - Elicit initial models and/or edit derived models
 - Analysis tool - Check models for completeness, errors, and calculate metrics
 - Transformation tool - Used to transform models into other models, code or documents.
 - Composition tool - Used to compose several source models, conforming to a "master" source model
 - Test tool - Used to test models
 - Simulation tool - Used to simulate the execution of a model
 - Metadata Management tool - Handle the general relations between different models, including metadata on each model and the mutual relations between these models
 - Reverse Engineering tool - Transforms particular legacy models or information into full-fledged models
 - Here's a link to the types of tools: http://en.wikipedia.org/wiki/Model-driven_architecture