

Panel MDA in Practice

Jean-Marc Jézéquel
IRISA (INRIA & Univ. Rennes 1)
Campus de Beaulieu
F-35042 Rennes, France
jezequel@irisa.fr

Wolfgang Emmerich
University College London
Dept. of Computer Science
Gower St, London WC1E 6BT, UK
w.emmerich@cs.ucl.ac.uk

On its web site (www.omg.org) the OMG gives its vision of MDA as follows:

MDA provides an open, vendor-neutral approach to the challenge of business and technology change. Based firmly upon OMG's established standards, MDA aims to separate business or application logic from underlying platform technology. Platform-independent applications built using MDA and associated standards can be realized on a range of open and proprietary platforms, including CORBA, J2EE, .NET, and Web Services or other Web-based platforms. Fully-specified platform-independent models (including behavior) can enable intellectual property to move away from technology-specific code, helping to insulate business applications from technology evolution, and further enable interoperability. In addition, business applications, freed from technology specifics, will be more able to evolve at the different pace of business evolution.

A somehow less enthusiastic perspective is given by Martin Fowler:

Some people think that Model Driven Architecture (MDA) will be the biggest shift in software development since the move from assembler to the first high level languages. Others think that it's nothing more than Night of the Living CASE Tools.
(martinfowler.com/bliki/ModelDrivenArchitecture.html)

Some of the ideas underlying MDA have indeed been around for many years, from the failed promises of CASE in the late 1980's to more recent approaches based on *wizards* or generative programming, or even Aspect Oriented Programming.

Because MDA gives a name and a conceptual framework to established practices, many people are realizing they were doing MDA without knowing it for many years,

even if it is not exactly *by the OMG's book*. It is not a surprise then that many vendors claim to produce products that support MDA, and that more and more success stories are being publicized (who will be the first one to dare write a MDA unleached book?).

This panel will try to help clear the dust by examining the technologies and tools behind Model Driven Architecture, and assessing their novelty and interest. Specifically each of the following questions will be addressed by a renowned expert in the field:

- What is MDA in practice?
- What is it useful for? On its Web site the OMG claims that the top three benefits of the MDA to enterprises trying to cope with today's computing environment are:
 - An architecture based on the MDA is always ready to deal with yesterday's, today's and tomorrow's "next big thing".
 - The MDA makes it easier to integrate applications and facilities across middleware boundaries.
 - Domain facilities defined in the MDA by OMG's Domain Task Forces will provide much wider interoperability by always being available on a domain's preferred platform, and on multiple platforms whenever there is a need.

How has it been correlated in the field?

- Is MDA more than Night of the Living CASE Tools?
- Are there new software engineering issues to consider with MDA?
- What is beyond the OMG's vision? How do other approaches such as Generative Programming or Aspect Oriented Software Development etc. fit to the overall picture of Model Driven System and Software Engineering?