

MDA, Thales, Nestle, and Microsoft

Adrian Kosmaczewski

2005-03-27

I haven't blogged for a while; I must admit that this has been a busy month. In fact until June I will be working inside one of the biggest companies on Earth: Nestle. My task, as a Software Architect, is to set up the overall architecture of a worldwide, distributed application, whose details I shall not lay down here (for I have signed a non-disclosure agreement). But I can say that this is, by far, the most exciting time of my professional life; this architecture is already the biggest challenge I have ever faced.

The application, following Nestle's alignment with Microsoft technologies, will be built using the .NET platform. It will integrate inside Nestle's Enterprise Service Bus, and thus will be using special SOAP services designed by Nestle's staff to handle several cross-cutting concerns such as security, reporting,

On the other hand, my employer, THALES, has become one of the members of the board of directors of the Object Management Group (OMG). This means that a global policy of THALES from now on will be to support one of the latest and biggest standards proposed from the OMG, that is, MDA, or Model-Driven Architecture. I will be blog more on MDA soon, since next April I will attend a special MDA workshop in THALES' headquarters in Paris.

I will not go deeper in the details of MDA, which you can find here, but I will nevertheless comment my particular situation right now. The problem is the following:

- Nestle has a strong commitment to using Microsoft based technologies; this in itself is fine, since I am mostly proficient with Microsoft's tools and technologies.
- Microsoft has a slightly different way of seeing application architecture and modelling, and it has definitely nothing to do with MDA.

I hope you see my point; maybe this article will help you see it easier:

Is that Model Driven Architecture (MDA)? No, not quite. Like MDA, we are interested in models. However, we are much less concerned with portability and platform independence than MDA, and much more concerned with productivity. While stereotypes and tags can be used to decorate UML sublanguages, experience suggests that

much more precise language features are required to support compilation, debugging, testing and other software development tasks. Unlike MDA, we do not propose to use UML for this purpose.

(...)

What about interoperability? Isn't MDA right in calling it a key piece of the story? Indeed, interoperability is a key piece of the story. In our view, however, it is much more important to standardize the protocols used to specify and assemble components, than the languages used to implement them. We think the web service protocols standardized by the World Wide Web Consortium (W3C) and the Web Services Interoperability Organization (WSI) hold much more promise for enabling interoperability than UML.

(...)

As you can see, Microsoft's view of things is quite different from the OMG's. Microsoft sees (and I recommend you a thorough reading of the above article) Software Factories and SOAP as the way to build the next generation Enterprise Service Bus, while MDA stresses modelling and portability; the key difference between both approaches has to do with the keywords "platform independence". Microsoft is not at all interested in that (of course not) and would rather bet on delivering productivity tools that help achieve faster time-to-market scores (which I believe is great). As soon as you stick to using Windows servers and Microsoft tools, you can get your apps up and running much faster than if you think "models".

This has raised a couple of questions in my head; these questions will be the main axis of my future work as an architect:

- What are the tradeoffs (if any) of both approaches? Or better still: how can I make them collaborate? How can I design this particular application using MDA? Is it worth it?
- How can I leverage THALES' commitment to MDA, and use this knowledge for building platform-specific solutions such as those needed by Microsoft-based organizations such as Nestle?
- Will the .NET platform grow as a standard such as J2EE? In this sense, will the CLI ECMA/ISO standardization process help? Will Mono provide an Enterprise-class platform, enabling portability and taking .NET development to a next big step towards other operating systems?
- Will Microsoft allow this to happen? I would not be that fast to say "NO"; Microsoft is going through deep internal power struggles, and it would not surprise me that, in the near future, the .NET and the Mono platforms would be backed together as a whole standard infrastructure platform (which I think will ultimately happen, since no closed systems can survive in today's market conditions). The J2EE approach (an open standard with several vendors providing implementations) is definitely interesting.

Waiting for the next big thing to happen, I will try to give answers to these questions in the near future.