A Response to Forrester

MDA Journal

April 2006

David S. Frankel
Lead Standards Architect - Model Driven Systems
SAP Labs

David.Frankel@SAP.com

https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/u/55914

I've noticed that a number of industry commentators do the following:

1) Seize upon a highly simplistic idea of what MDA is
2) Claim that that simplistic idea is what MDA is all about
3) Conclude, therefore, that MDA is bad

The recent Forrester article, "MDA is DOA, Partly Thanks to SOA,"[1] is a case in point. The article's distortions of MDA are particularly drastic, and, thus, I feel compelled to respond.

Code Generation Extremism

The article is permeated with the allegation that MDA requires fully automatic code generation, and that any approach that mixes automatic code generation with manual code writing is "pragmatic MDA," but not "OMG MDA." On that basis, the article denounces MDA as impractical and dead. It cites Compuware's OptimalJ tool, which is one of the tools that is the most truly compliant with MDA standards, as being typical of the pragmatic MDA but not OMG-MDA.

To see the weakness of this claim about MDA's attitude toward code generation, one has only to look at the OMG's official *MDA Guide*, published in 2003.[2] The chapter named *MDA Transformations* contains Section 4.1 called *Degrees and Methods of Model Transformation*, which starts as follows: "There is a range of tool support for model transformation. Transformations can use different mixtures of manual and automatic transformation." The section goes on to outline four kinds of MDA transformations:

1) Manual transformation
2) Transforming a PIM Prepared Using a Profile
3) Transformation Using Patterns and Markings
4) Automatic Transformation

So, according to the MDA Guide, automatic transformation is but one of four kinds of MDA transformations. Most of the major books on MDA also discuss hybrid approaches involving a mix of generated and manually written code.[3] Mellor, et al., explicitly discuss how to manage such mixtures[4], and my first book has detailed descriptions of such scenarios as well.[5]

Whose Myths?

The Forrester article cites two myths that it claims MDA proponents believe:

> Myth 1) "It's possible to model systems without regard to implementation details."

It's possible to write Java code without regard to the architecture of the CPU, isn't it? It's all a matter of degree. But what the article is really implying comes out later when it says, "MDA is predicated on an assumption that the implementation platform doesn't matter..."

MDA Journal

April 2006

Anyone who thinks that is what MDA is about certainly hasn't delved into the works of the leading MDA thinkers I cited above. Many of us have published and spoken extensively about various approaches to taking abstract and implementation considerations into account in using MDA, with a far more sophisticated point of view than Forrester's caricature would suggest.

MDA seeks to *separate* concerns about implementation platforms from other concerns, not ignore them. The various writings and debates within the MDA community reveal a good deal of soul searching about the trade offs among different means of capturing implementation concerns, including model annotations, model compilers, and manually-written code. But everyone agrees that implementation platforms are relevant, and it's a gross misrepresentation to suggest otherwise.

Myth 2) "It's possible to transform systems without human intervention."

My citations above, from the MDA Guide about model transformations and from other MDA publications about mixtures of generated and manually written code, should lay to rest the accusation that MDA makes sweeping claims in this regard. There are some cases where full code generation is possible, and there are some cases where it isn't.

The real myth is that MDA proponents believe and promote these two myths.

Waterfall or Iterative?

The Forrester article says that "MDA is inherently waterfall."

I know of no prominent MDA thinker who treats MDA in a waterfall fashion. I state flatly in my first book — one of the first books to define MDA comprehensively — that MDA is iterative rather than waterfall, and I demonstrate several different kinds of iterative workflows that MDA practitioners use.[6]

Kleppe, et al., also make it clear, early in their discussion, that the MDA software development life cycle is iterative.[7] Mellor, et al., in discussing a model-driven development process, state, "The overall structure of the process, then, is not only iterative and incremental, but recursive."[8]

Monolithic or Componentized?

The article also says in so many words that MDA embraces monolithic rather than componentized systems.

This again runs counter to what the MDA experts have been saying. My book makes it clear that MDA has to leverage what the industry has learned about component-based development. Many of the examples in the book are openly based on the component-based architecture described in Peter Herzum and Oliver Sims' seminal book *Business Component Factory*.[9]
As for my colleagues' works, rather than citing chapter and verse in this case, let me just say that these people are some of the most mature and talented

MDA Journal

April 2006

architects on the planet, fundamentally grounded in concepts of modularity and flexibility, truly at opposite poles from how they are being portrayed.

Where's the Metadata?

It's also interesting that the Forrester article evidences no concept of the core role that metadata management plays in MDA. It doesn't mention the fact that Eclipse's metadata management makes heavy use of XMI®, one of the key MDA standards. Eclipse has tremendous traction, and hundreds of tools are being built upon its metadata management infrastructure. Eclipse doesn't align completely with all of the MDA metadata management standards, but its implementation of XMI does.

MDA vs. SOA: A False Choice

An underlying premise of the article is that SOA is better than MDA because SOA is pragmatic, iterative, and componentized, while MDA is rigid, waterfall, and monolithic.

I've already dealt with the false charges about MDA in this regard. Any competent architect practicing SOA or MDA would use a flexible, iterative, component-based approach.

But it's also worth mentioning that there's no contradiction between SOA and MDA. If you want to practice SOA in a manner that raises the level of abstraction so that as much low-level, mechanistic code as possible is taken outside the sphere of the application programmer, and if you want to practice it with a consistent approach to managing the different kinds of metadata that SOA needs at various levels of abstraction, then MDA can be helpful.

In fact, the need to get a handle on the complexity of modern-day service-oriented systems is one of the things that are impelling the industry to try to find ways to simplify development, leading to initiatives such as MDA and Microsoft's Software Factories. Both of these initiatives are still in their early stages and are part of a long-term transition. As I've written repeatedly, neither solves all problems, and it will still take a number of years before the tools and standards reach a mature state, which is true of SOA as well.

The Straw Man Fallacy

I think I've established that the article seriously mischaracterizes MDA and what its leading proponents have been saying. Forrester clearly hasn't done its homework on this subject. The article sets up a red herring that has no possibility of viable life and then pronounces the fish dead.

This kind of belittling non-reasoning is sometimes called "the straw man fallacy." Here is a succinct description of the straw man fallacy:[10]

MDA Journal

April 2006

The Straw Man fallacy is committed when a person simply ignores a person's actual position and substitutes a distorted, exaggerated, or misrepresented version of that position. This sort of "reasoning" has the following pattern:

1. Person A has position X.
2. Person B presents position Y (which is a distorted version of X).
3. Person B attacks position Y.
4. Therefore X is false/incorrect/flawed.

I was becoming alarmed about a tendency to pigeonhole MDA into a shallow and simplistic corner a couple of years ago, despite the publications that explained MDA in a more nuanced fashion. I wrote an article about this problem, which appeared as my MDA Journal for March 2004[11].

In that article I was sympathetic to those who were misconstruing MDA. I wanted to give them the benefit of the doubt. Marketing literature tends to hype, the OMG's MDA home page leaves a lot to be desired, and at the time the major publications about MDA were still quite new. But this is 2006. There's no excuse for this level of ignorance on the part of a major analyst mouthpiece, which has had plenty of time to research the subject, and has an obligation to dig deeper than surface marketing missives.

I'm not here to defend everything about MDA. Some of the MDA standards that have been adopted recently could be better. I'm sure that my ideas could benefit from scrutiny and honest debate. But I object to putting words in my mouth and the mouths of my colleagues, words that we made clear from the start are not what MDA is about.

—David Frankel

Footnotes

[1] Forrester, March 22, 2006.
[2] www.omg.org/docs/omg/03-06-01
[3] See, for example, the following major MDA publications. Each of these publications has a forward by a senior executive of the OMG:
• Stephen J. Mellor, Kendall Scott, Axel Uhl, Dir Weise, *MDA Distilled: Principles of Model-Driven Architecture*, Addison-Wesley, 2004.
• Anneke Kleppe, Jos Warmer, Wim Bast, *MDA Explained: The Model Driven Architecture Practice and Promise*, Addison-Wesley 2003.
• David S. Frankel, *Model-Driven Architecture: Applying MDA to Enterprise Computing*, John Wiley and Sons, OMG Press, 2003.
[4] See, for example, Mellor et al., Chapter 8, Section entitled "Managing Manual Changes to Generated Models," pp. 85-87. (The authors use the term "model" expansively to include code).
[5] Frankel, Chapter 8, Section entitled "Synchronizing Models and Code," pp. 230-244.
[6] Ibid.
[7] Kleppe et al, Section 1.2.1 (see Figure 1-2 on page 3 for a schematic that is obviously iterative).
[8] Mellor, et al., page 125.

MDA Journal

April 2006

[9] Peter Herzum and Oliver Sims, *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*, John Wiley and Sons OMG Press, 2000.
[10]The Nizkor Project, http://www.nizkor.org/features/fallacies/straw-man.html
[11] "The MDA Marketing Message and the MDA Reality," MDA Journal, Business Process Trends, March 2004. http://www.bptrends.com/search.cfm?keyword=MDA+%22Marketing+Message%22&gogo=1&go.x=107&go.y=9