

Executive Justification for Adopting Model Driven Architecture (MDA)

Stanley J. Sewall

Away from the Hype...MDA gives companies a viable alternative to application development instead of corporate stagnation or offshore

One of the key things that differentiates model driven application development from conventional enterprise application development is the emphasis model driven development places on design. With the Model Driven Architecture (MDA) approach, you construct only the most important aspects of the enterprise in the application, and this saves time and money. In this paper I want to assess the direct payback users achieve, in terms of increased revenues or profits, and reduced operating costs that results from increasing the design effort. Ultimately, I suggest, the adoption of MDA will be driven by the superior payback achieved by MDA technologies.

Key Reasons For Adopting MDA

Companies adopt MDA for a variety of reasons. Some adopt MDA to achieve long-term strategic competitive advantage while others adopt MDA to achieve immediate business results. Most companies have application data management systems in place, some performing in a satisfactory manner and some not. All companies need to be able to integrate data into their legacy systems in a timely manner. Failure to do so could manifest itself in acquiring the wrong component and that could ultimately affect shipments to their customers, or result in costly internally corrective actions.

Many companies have become "virtualized" software houses, depending heavily on outsourced resources for tactical tasks that need to be done in a short amount of time. Thus, executives need to put their own organizations in order first, before focusing on key development endeavors for other companies.

Some organizations are unable to launch any type of IT endeavor because of concerns like the following:

- Culture
- Politics
- Fear of Technology
- Credibility
- Strategic Pressures
- Cutbacks

In the case of MDA endeavors, there is a fear of the new technology and what it will expose about the organization's current infrastructure. Most of a company's application business logic is concealed in scattered documentation, or locked up in the heads of various company employees. Some people use this knowledge to wield a big stick in the organization and gain visibility within the company. In essence, managers are potentially threatened to being held hostage by their IT employees who have private and separate familiarity with the company's computer infrastructure.

Most companies' business models force their IT personnel to function in reactive mode, with very little time and energy left for design. Organizations have to react quickly to market dynamics, leaving very little time to focus on design, due to the lack of value placed on the design phase of the project. It is during the design phase that a business focuses attention to its business logic.

In addition to the devaluing of design generally, there is a dearth of object modeling talent accessible to brick-and-mortar companies to help them through the design phase. Documentation specialists are often the first causalities during a company's financial cutbacks.

MDA application development addresses a number of problems currently facing C-level executives, including:

- Application Development Backlog
- Limited Budget
- Legacy Staff
- Credibility
- Linking Applications to Business Goals

Developers may find that the systems they have inherited provide very little in terms of design specification. Often times, contractors are hired who lack an awareness of external processes that will effect the system they are being asked to change. One of the benefits of MDA is that there is a direct alignment between design and the actual application code.

One of MDA's strengths lies in separating the business logic of the company from the technology infrastructure. By adopting this approach, companies can rapidly adapt to changing technology landscapes, like wireless or EAI deployments, without having to invest heavily in a great number of developers and operational personnel to support the new technology. Thus, because the business logic is encapsulated in UML, it can be applied to different technology deployments like J2EE or .Net, resulting in a usable source code. This provides a strategic competitive advantage for a company that has to react quickly.

The key to any MDA development project is UML. By utilizing UML, the MDA application development lifecycle is reduced in the following areas:

Activity	% of Savings
Business Requirements Gathering	10%
Design and UML Modeling	-20%
Application Construction	60%
Application Iterations	70%
Testing	80%
Documentation	70%
Runtime Environment Tuning	85%
Project Management	60%

In addition, the UML diagrams become a vehicle for communication with the other business units. I have had tremendous success with incorporating functionality through supporting documentation from the operational business units. Compliance with other business units is easier because of the ability to design, document, develop, and deploy applications from the business units' requests.

It should be noted that MDA, as well as BPM (an off-shoot of MDA), should not be given to the business analysts or subject matter experts with the expectation that they will develop workable source code. MDA relies on communication from the business units to the UML Data Modelers to interpret the business process in the form of a UML model. IT personnel deploy the model to the business domain.

ROI - Strategic and Tactical Benefits

In this section I shall contrast strategic benefits that impact long term success with tactical or operational benefits aimed at producing a short-term payback. In tough times, the latter tend to weigh more heavily in project initiation, whereas, in good times, strategic benefits get more play with business users, always assuming the benefits are tangible. I believe that all payback, whether it bears fruit in the long or short term, should be measurable in some objective or commonly-agreed upon manner. Thus, all such benefits should be considered "tangible" and admissible in an objective ROI assessment.

In fact, I would argue that it is *always* better for ROI cases to be based on both strategic and tactical/operational payback. This balance makes for a lower risk of frustration in the event that either one of the two results fall short of expectations. For example, if you set out to fix a broken part of the business that has demonstrable results of far-reaching, or strategic, value, but don't actually fix the entire process, you may still have "won," if you achieve, say, a 30% or higher gain in productivity. Besides the anticipated benefits that may figure in customers' initial ROI projections, it is worth noting that in many cases there are *unintended consequences* that can actually count for more in the long run, thus strengthening the case for ongoing or increased deployment of the solution. You will hear more about this below.

By implementing a single point of responsibility for the software development approach within its IT development infrastructure, a company can launch several development application endeavors with an average development cycle of three to four months, if it uses a strong modeling methodology like MDA. Because MDA can generate a large percentage of J2EE/.Net applications from a model, the speed of development and financial exposure to changes in implementation on a project is greatly reduced. Thus, it is much more than a mere question of financial ROI; it is a basic survival strategy in their ultra-competitive market.

Key Strategic Questions

People frequently ask me how the current downturn has impacted the consulting business, and my customary answer has been: "Well, fortunately, most companies find MDA attractive." Due to the dramatic nature of many companies' problems, today, no matter whether you are talking about public or private software companies, corporate strategy matters much more today than it seemed to matter during the halcyon days of the 1998-2000 Internet bubble. The main difference I see today is in the level of urgency of many companies' predicaments and the serious consequences of staying with the existing technology strategy or picking a wrong development strategy.

When you look at the cost of MDA versus what I call "traditional development," use see that MDA will reduce the cost of application development by 40 – 50%. The

savings occur whether you are designing systems integration, designing B2B marketplaces, or undertaking other custom application development. Because of the reduction of J2EE/.Net application costs, companies who venture into the MDA development will experience significant cost savings for application development. In my travels, in talking with C-level executives, I found they all acknowledged the importance of having the design actually match the application code. For the first time in our development arena, source code generation is viable and also customizable.

The Object Management Group (<http://www.omg.org>) is a standards group that has defined the standards for the MDA technology.

Business advantage occurs when you allow the IT personnel to view the software application framework in a federated manner. Today, many IT developers are confused by the latest trends of application development and lose sight of what provides true business value for their organizations. Many organizations I have worked with need to change their development practices so significantly that the entire mission of the company requires a fresh analysis and a rework. During the course of working with such organizations, I have refined a set of questions designed to guide customers through a thorough review and, if necessary, an overhaul of their business strategy and IT operations. The questions extend beyond IT strategy when considering operating strategy, though a focus on the market is invariably the starting point.

Here are the six key strategic questions we have been using:

1. Have you been able to meet your customer's demands? The current technology fad of offshore development does not fundamentally solve the problem of the disjointed application development. MDA solves this problem with demonstrable results of design and an actual working IT system.
2. What application paradigm are you in today? How "investable" is it - or can it become - over time? Are you in more than one category, and are there any new, more powerful categories that you could credibly enter to gain a competitive advantage?
3. How powerful a POSITION has your company achieved (or can it achieve) with your development practices and with meeting market demand within present and future category(ies)? How can you establish a differentiated position in the most powerful category? Alternatively, should you consider participating in a different category in order to achieve more power in the market?
4. Can your existing IT personnel adapt to a new development paradigm? I have found more senior IT workers generally gravitate toward using MDA. Over their long careers, they know the importance of design and documentation to save time, financial resources, and maintenance over the lifetime of the application.
5. How can the company OPTIMIZE RESOURCES in order to maximize competitive advantage? How can you shed contextual (i.e., non-core)

activities in order to focus today's resources (time, talent, and management attention) on core operational activities?

6. What is your CHANGE MANAGEMENT plan, and who is leading it? Is this individual and their [or "his or her"] team sufficiently committed and authorized to achieve their objective?

Wrap Up

Today's brutal environment for prioritizing IT investments in large enterprises is causing many decision cycles to be either significantly extended, or canceled. The immediate consequence, for IT groups, is an increased pressure to deliver what operations and marketing groups are demanding. Managers and IT project leaders are anxious to change, since they want to avoid cost overruns and the difficulties they have encountered on past projects. Unfortunately, change is also fraught with anxiety, centered on the fear of trying something different.

When all is said and done, my recommendation to companies who are thinking of using MDA is to identify an addressable target project where you can gain a sizable and demonstrable ROI. Remember you still have to design anyway! The key task is to prioritize your attack around a segment with a broken business process that you are uniquely capable of fixing. Around this "compelling reason," there are **at least three value-building activities** that are pragmatic in nature to the value of a company:

- Identify one to three business units that are willing to make an investment with the IT organization to help develop the application via business object modeling.
- Continuously maintain the discipline of object modeling, remembering the importance of developing the application without reverting to the old ways of application development.
- Hire or contract an experienced IT person with Object Modeling. MDA relies heavily on UML and on experience with analysis and design. Hiring a person experienced with MDA, who will help the company transition to an MDA paradigm, is a small investment that pays big returns.

Stan Sewall is an enterprise level consultant. He can be contacted at:
stan.sewall@juno.com or at (+1) 404.307.2643.