

# KEEPING YOUR BUSINESS RELEVANT WITH MODEL DRIVEN ARCHITECTURE® (MDA®)



## THE BORLAND ADVANTAGE

MODEL DRIVEN ARCHITECTURE® (MDA®), TOUTED AS THE BIGGEST SHIFT IN SOFTWARE DEVELOPMENT SINCE THE MOVE FROM THE ASSEMBLER, AIMS TO HELP ORGANIZATIONS PROTECT AND MAINTAIN THEIR INVESTMENTS IN SOFTWARE. A LEADER IN MODEL DRIVEN DEVELOPMENT,™ (MDD™) BORLAND EXPLORES THE PROMISE OF MDA, THE CHALLENGES IT PRESENTS, AND YOUR PATH TO SUCCESS.

## Contents

- Executive summary** .....2
- What is MDA?** .....2
- The promise of MDA** .....2
- Business drivers for MDA** .....4
- The challenges of MDA** .....5
  - Maturity of MDA standards* .....5
  - The Complexity of UML: The need for the “expert modeler” skill-set* .....5
  - MDA can only do so much* .....5
- The Borland Advantage** .....6
  - 1. Eased learning curve and team collaboration* .....6
  - 2. Standards* .....6
  - 3. Automation* .....7
  - 4. Market Leadership* .....7
  - 5. Business-optimized software* .....8
- Eye on the future: Conclusion** .....8

## Executive summary

If you are reading this overview, you and your organization are probably seeking better, more efficient ways to secure software investments. Predicting technology trends is not your business, but the flexibility to adapt to them is. No one wants their expensive, mission-critical software applications to be pushed into obsolescence, yet evolving technology, combined with changing business needs, greatly increases the risk of this unwanted outcome.

In response to today's fast-paced, competitive climate, the platforms and infrastructure that you have invested in are constantly changing. If applications cannot be adapted to platform or infrastructure changes, they become redundant. Technology advances aside, the need for software that scales to changing business needs is stronger than ever and is now a necessity rather than a luxury.

To remain competitive and relevant to customers, organizations strive to become more agile. If companies want to use IT to enable future business strategy and to capitalize on emerging technologies and trends, they must first assess their needs and then make the necessary technology investments according to those assessments.

### What is MDA?

Model Driven Architecture (MDA), developed by the Object Management Group™ (OMG™), recognizes that to take advantage of new technologies for competitive advantage, applications can be made independent of the infrastructures they use. MDA was designed to help organizations rapidly adopt new technologies and concepts without necessitating a rewrite of their entire systems.

In this paper, we explore the promise of MDA, the challenges of adopting an MDA approach, and “The Borland Advantage”—how Borland can help you today.

### The promise of MDA

MDA is touted as the biggest shift in software development since the move from assembler machine code to the first high-level languages. With a set of software standards that impose discipline in the development process, MDA aims to increase your control over the applications you have invested in and also over the changes that affect your business.

At the heart of MDA is the OMG standard, the Unified Modeling Language.™ (UML™), which uses diagrams—“models”—to design and describe software applications. Using models as an essential part of the development process is now recognized as a best practice, known generally as Model Driven Development (MDD).

MDA, however, attempts to broaden the benefits of using models with model-to-model transformation—the ability to generate entire applications without writing a single line of code. The OMG ([www.omg.org](http://www.omg.org)) offers examples of organizations that have generated 100% of their code through MDA by combining modeling with model transformation.

To ensure that applications do not become obsolete as platforms and infrastructures evolve, MDA demands that software be defined independently of the infrastructure or platform on which it is executed. This goal is achieved by creating Platform-Independent Models (PIMs).

However, in order for the software to then take advantage of a specific platform or infrastructure, PIMs must be transformed to Platform-Specific Models (PSMs), which describe the application within the native constructs of the target platform. MDA is targeted at businesses that realize the importance of keeping their software development relevant in the ever-evolving marketplace.

## Business drivers for MDA

<b>Business driver</b>	<b>MDA promise</b>
Reduced risk of lock-in to specific technologies	Ensures that rapid changes in technology frameworks do not render applications useless
Preservation of investment in application development	As above, ensures that rapid changes in technology frameworks do not render applications useless
Increased productivity	Through promotion of code reuse and code generation, repetitive coding is eliminated from projects
Increased business agility	Ability to respond to changing business needs by changing the application model to fit changes at requirements level
Reduced development and maintenance costs and reduced time-to-market	<ul style="list-style-type: none"><li>• Simplifies the task of maintaining the software during a long production lifetime by being able to autoport the application to new platforms</li><li>• Maintenance activities rendered more cost-effective</li><li>• Manpower overhead required to create software reduced</li><li>• No need to code complex infrastructure plumbing</li><li>• Automation of repetitive development tasks</li></ul>
Improved application quality	<ul style="list-style-type: none"><li>• MDA recognizes the need to improve source code: the less code written, the less liability assumed</li><li>• Tested software blueprints, industry-standard patterns, can be automatically applied. This prevents the deviations from architecture and design guidelines that are the primary source of scalability, performance, and availability problems.</li></ul>
Software that meets business needs	<ul style="list-style-type: none"><li>• Business-focussed approach to software development, with the ability to quickly react to changing business conditions</li><li>• Tested application blueprints help provide industry-standard software</li></ul>

## The challenges of MDA

### Maturity of MDA standards

For maximum benefit and increased adoption, the challenges of MDA should be considered. Because the original UML standard was computationally incomplete, the software industry is pinning its hopes on UML 2.0 and Object constraint Language (OCL) 2, which have been adopted but are not finalized. Other standards considered important for MDA, such as a Java™ implementation for MOF™ 2 Java Metadata Interface (JMI), and a model transformation standard (QVT), have been neither adopted nor finalized.<sup>1</sup> Even before the standards are finalized, vendors lobby to have their solutions recognized as a standard, sometimes resulting in proprietary implementations that can weaken the promise of MDA.

### The complexity of UML: The need for the “expert modeler” skill-set

The technical complexity of UML has been held responsible for modeling adoption issues. Few expert modelers can rapidly evolve an application from requirements to code. Although the benefits outweigh the drawbacks, skill-set issues can adversely affect time-to-market. Many of today’s modelers are casual in their approach; MDA, however, requires increased rigor and training in UML modeling.

### MDA can only do so much

It is unrealistic to expect 100% code generation for every computing problem, and no vendor in today’s market can realistically offer a complete MDA solution. If you expect too much of MDA, it will fail. It merely offers another layer of reusable technology and abstraction designed to integrate with the development tools you already know.

To realize the full benefits of MDA, organizations must support the full software lifecycle development process, from analysis and requirements management through design, development, implementation, deployment, and maintenance.

---

<sup>1</sup> At the time of this paper (May 2004)

## The Borland Advantage

Among the last true independent software vendors, Borland offers a pragmatic, non-proprietary view that affords customers the freedom to move into the future without abandoning the past. As co-chair of the Object Management Group™ (OMG™) subcommittee in charge of the MDA Guide and an active participant in UML™ 2.0 and OCL 2 standards work, Borland's involvement with MDA runs deep. Borland has a clear path set to get organizations on board with MDA today and has established recognized leadership regarding many concepts MDA brings to the mainstream, especially in Model Driven Development (MDD) and model transformation and patterns (application blueprints). Here are just some of the ways Borland can help you today.

### 1. Eased learning curve and team collaboration

Borland is focused on making the visualization of your software through UML modeling as pain-free as possible. A major barrier to MDA adoption is the requirement that UML must be learned and used. Borland is a leader in working to lower this barrier by addressing the human and technology obstacles to adoption. Borland® LiveSource™ technology, for example, ensures that models and code are in synch, providing greater levels of collaboration between the architects and developers in your organization. Furthermore, Borland® Together® modeling tools allow you the flexibility to choose your version control system. Other modeling tools introduce conflicts between their own repository and your choice for configuration management.

### 2. Standards

Borland is committed to following standards rather than introducing proprietary methods. Borland is focused on helping organizations leverage the advantages of key MDA standards, including XML Metadata Interchange (XMI), Object Constraint Language (OCL), and UML within its technologies. We are a leading contributor to evolving the UML 2.0 specification and JMI for MOF 2. While standards for model transformation (QVT) are in the early stages, Borland has led the charge with its pattern transformation technology.

### 3. Automation

Borland provides a high degree of automation. Borland® Enterprise Core Objects (ECO™) is a model-driven runtime platform that automates many of the manual steps involved in an MDA approach to development, with support for UML, OCL, and XMI.® ECO technology, designed to automate the development of business applications and broadly used in mission-critical environments, offers specific application services such as persistence, transactions, versioning, and integrity management.

Borland® Together® ControlCenter™ facilitates the automation of model transformation for data, application models, and patterns. This transformation provides the flexibility to target database servers, programming languages, and application servers.

### 4. Market leadership

Borland ECO offers the first Rapid MDA technology for the Microsoft® .NET Framework. Borland Together technologies, launched in 1996 by TogetherSoft, originated the idea that a model could be more than pictures and could significantly contribute to software development. Borland® Together® technologies include: Borland® Together® Edition for Eclipse, Borland Together Edition for Microsoft® Visual Studio® .NET, and Borland Together ControlCenter.

Borland Together technologies—first to market in 2000 with award-winning design pattern transformation technology—enable organizations to leverage industry-standard application blueprints and to generate related application models and code.

Supporting PIM and PSM, Borland is recognized by analyst firms, including Gartner, as a leader in UML-based Model Driven Development. Borland offers true cross-platform MDA—the ability for PIMs to be deployed to multiple platforms from within one product, Together ControlCenter. This provides language-to-model integration through Borland® LiveSource™ technology, and language-independent modeling by allowing modeling in a design-only language. With a design language, ControlCenter is able to allow for the modeling of an application regardless of the underlying platform language. These models are then transferred to a platform-specific model, with the desired language source generated in the process.

## 5. Business-optimized software

Borland's mature, disciplined approach to development produces business-optimized software. As the leading independent vendor of standards-based, cross-platform, integrated application lifecycle management solutions, we offer an end-to-end solution—from analysis and requirements management through design, implementation, deployment, and maintenance—helping to mitigate the risk of failed projects.

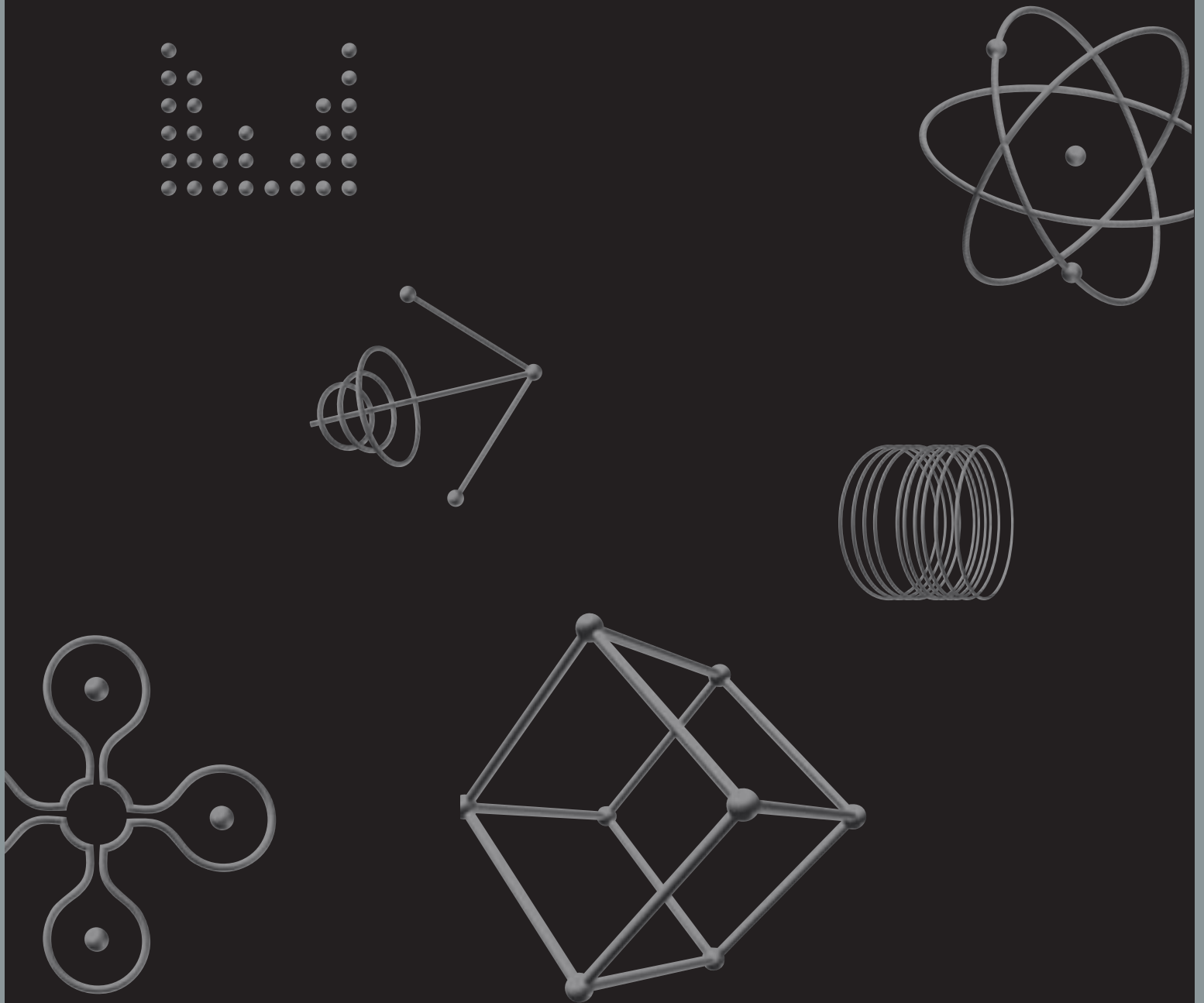
## Eye on the future: Conclusion

The many concepts and drivers behind MDA are solid and well understood: to protect and maintain investments in software through integrating modeling into the development process. Describing solutions in abstract terms and mapping them to platform-specific implementations gives you the freedom and agility to adopt new technologies for competitive advantage. However, the lack of maturity of some MDA standards and specifications are prohibitive to organizations interested in taking full advantage of all that MDA offers. Until these standards and specifications evolve, silver-bullet solutions for MDA do not exist.

Borland, with its history of embracing the many key concepts at the heart of MDA, is uniquely positioned to offer a leading, safe, market-proven, pragmatic implementation of MDA for maximum benefit and increased adoption within your organization.

With enhanced integration of modeling with software development lifecycle stages and planned innovations that advance the reuse of industry-standard patterns, Borland reduces code liability through new model-to-model transformation technology. In doing so, Borland helps drive business agility, in turn keeping your development relevant for the long term.

Made in Borland® Copyright © 2004 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. CORBA and ORB are trademarks or registered trademarks of Object Management Group, Inc. Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Microsoft, Windows, and other Microsoft product names are trademarks or registered trademarks of Microsoft Corporation in the U.S. and other countries. All other marks are the property of their respective owners. Corporate Headquarters: 100 Enterprise Way, Scotts Valley, CA 95066-3249 • 831-431-1000 • www.borland.com • Offices in: Australia, Brazil, Canada, China, Czech Republic, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Italy, Japan, Korea, Mexico, the Netherlands, New Zealand, Russia, Singapore, Spain, Sweden, Taiwan, the United Kingdom, and the United States. • 22220.1



## About Borland

Borland is the world leader in Java™ and cross-platform development solutions. We have one of the largest business alliances in development environments for the Microsoft® .NET and Windows® platforms. Squarely focused on improving customers' development and deployment capabilities, Borland continues this tradition of excellence with technologies designed to accelerate the entire application lifecycle. Borland is recognized by industry and financial analysts as one of the premier companies in advancing development technology. Fiercely independent for over 20 years and dedicated to producing excellent technologies, Borland is committed to helping companies create sustainable competitive advantage with custom software. Today's Borland, just like yesterday's, only faster, yet.

**Borland®**  
Excellence Endures™

100 Enterprise Way  
Scotts Valley, California 95066-3249  
Tel. 831-431-1000 [www.borland.com](http://www.borland.com)

Made in Borland® Copyright © 2004 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. CORBA and ORB are trademarks or registered trademarks of Object Management Group, Inc. in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other marks are the property of their respective owners. Corporate Headquarters: 100 Enterprise Way, Scotts Valley, CA 95066-3249 • 831-431-1000 • [www.borland.com](http://www.borland.com) • Offices in: Australia, Brazil, Canada, China, Czech Republic, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Italy, Japan, Korea, Mexico, the Netherlands, New Zealand, Russia, Singapore, Spain, Sweden, Taiwan, the United Kingdom, and the United States. • 22220.1