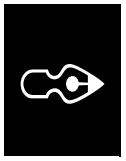

Foreword

"The sooner you start, the longer it takes." This paradoxical slogan was coined by Fred Brooks in 1975 in his seminal software engineering text, *The Mythical Man Month*, to emphasize that in software projects preparation is never wasted. If you skimp on requirements capture, you'll waste effort designing things the customer doesn't want. If you skimp on design, you'll write a lot of pointless code that doesn't solve his problem . . . and so on.

Dr. Brooks' dictum reminds us that Good Design Matters. It's just as true now as it was twenty-five years ago, but much else has changed. His book was based on his experience with OS/360, at that time one of the largest software projects ever undertaken. Since then, software size and complexity have grown beyond recognition, as has the bewildering choice of implementation technologies. Should we be using C++, Java, Visual Basic, or C#? CORBA? .NET? Web services? ebXML? EJB? JavaScript? ASP? JSP? SQL? ODBC? JDBC? Today, a single software project typically uses several of these, placing each where its particular strengths are best used. We have also become increasingly aware of the importance of software maintenance. Mission-critical software can continue in use for decades, suffering constant upgrades to cope with shifting requirements and changing technologies, so that in time maintenance costs exceed all others combined.

This book describes the Model Driven Architecture (MDA) approach to creating good designs that cope with multiple-implementation technologies and extended software lifetimes. MDA is based on widely-used industry standards for visualizing, storing, and exchanging software designs and models. The best-known of these standards is the Unified Modeling Language (UML). The Object Management Group's (OMG) creation of UML has promoted good design by providing a common, widely-understood visual language for engineers to exchange and document their ideas, and this has led to a dramatic increase in the use of visual modeling. However, visual modeling has too often been seen merely as a way to draw pictures of software, pictures that must later be laboriously translated into runnable code. One of the traditional excuses for skimping on design is that comprehensive models are "just paper," and effort spent creating them could better be spent on writing real code. Partly as a result of this,

there's a trend today towards development techniques that emphasize creating executable code instead of "mere" designs.

In contrast, MDA heavily emphasizes creating designs—not paper designs, but machine-readable models stored in standardized repositories. The intellectual effort invested in these models doesn't just sit passively on the page waiting to be laboriously recast as code. Instead, MDA models are understood by automatic tools from multiple vendors that generate schemas, code skeletons, test harnesses, integration code, and deployment scripts for the multiple platforms used in a typical project. Design effort invested in MDA models is repeatedly reused to generate multiple components, and by being updated over the life of an application, provides accurate documentation of how much-maintained software really works, rather than a frozen image of how things looked at the end of the design phase. In short, MDA is an architecture for creating good designs in today's multiplatform IT environment.

The authors of this book are well qualified to describe and document the MDA vision. They have contributed to OMG's creation of MDA, worked on commercial products that implement it, and used it in practice. The book is founded on solid theory, but it is nevertheless a practitioners' handbook, built around real-world examples, and offering guidance to IT professionals facing the ever-present problems of bringing in next year's project on time and on budget. Time spent reading it will not be time wasted. Remember: the sooner you start, the longer it takes.

Andrew Watson
OMG Vice President & Technical Director
35,000 ft. over Greenland
November 17th, 2002