Foreword

When the public talks about Artificial Intelligence, in most cases they actually think it still belongs to the science-fiction realm. The great Alan Turing predicted in 1950 that in about fifty years, "an average interrogator, after five minutes of questioning, will not have more than a 70% chance of identifying whether or not a computer is indeed a computer." However, the enthusiasm for a real "thinking machine" has slowly fizzled, and we are now taking incremental, evolutionary steps toward "smarter computers." Researchers are now tackling smaller problems than the thinking machine, but it doesn't mean that AI research hasn't produced real-world results. Neural networks technology has been used for a wide variety of tasks, from stock market prediction to computer games; speech recognition is now entering mainstream; various machine translation services are available online; handwriting recognition, as well as optical character recognition, can now translate arbitrary (type) written script into text.

Being involved in various AI-related development projects for almost 15 years, I did my share of programming in Prolog, LISP, Smalltalk, C, Java, and similar languages and development environments. Judging from my previous experience, the .NET framework and ASP.NET represent an exciting and important evolutionary step in the software development world. They provide a feature-rich application and development environment and are language and platform neutral—although more efforts will have to be invested to achive full platform independency through projects such as Mono. It is all about reducing the burden and the complexity of software creation. It is also about winning the hearts of developers all over the world through the support for collaborative projects, workspaces, and even the open-source movement, something that Microsoft has criticized

strongly at times in the past. .NET does require a shift in developer mentality, but the time invested in it quickly pays off.

So it is time to put these two, Artificial Intelligence and .NET, together. Both technologies are innovative; both are ambitious. Considering the amount of hype surrounding almost all AI-related technologies, it should be easy to find your way through, right? Think again: .NET is a newcomer in this area, and don't expect to find plenty of resources on AI development using its tools and languages. Most code samples and tutorials are written using traditional AI tools, and converting them to .NET can be a slow and painstaking process. This is where Sara Rea fills a huge gap with this book and its practical, hands-on approach. The four AI-related areas she covered offer unlimited potential for business-related applications. After all, a typical professional software developer doesn't have time to reinvent the wheel and needs a focused and pragamatic guide to this vast field. From speech applications to data mining, rule-based systems and agents, this book offers a solid instructional text of basic theory, the principles from which it derives, and how it is practically applied to develop advanced products.

As I was witnessing this book being written, I quickly discovered Sara's passion for knowledge and a talent in making hard things easy. I'm sure you'll appreciate that many advanced topics are presented in an easy-to-digest form thanks to this talent. Unlike many other technical books, it will not leave you feeling overwhelmed, but motivated to apply the techniques described inside to the real-world problems. And that's what software development is all about: building better and smarter solutions. Only this time, they'll be really smart.

Denis Susac
CEO of Mono Software,
http://www.mono-software.com

Preface

Artificial Intelligence (AI) has been around almost as long as computers. It was first introduced in the 1950s, and many scientists soon developed unrealistic expectations surrounding it. Throughout the past fifty years there have been as many advances as disappointments. Most people do not realize that AI-based technologies are being utilized every day. This is because once a technology becomes mainstream, it is generally no longer considered to be AI-related.

Take the things we associate with intelligence—speaking, writing, thinking; these are actually some of the easiest forms for AI to reproduce. The real beauty of our intelligence is that, while not easily seen, it exists deep inside every living cell that performs a specific function. No individual cell is smart enough to function alone. Instead, humans are a complicated network of cells that carry the blueprints for all other cells.

AI is not an all-or-nothing situation. It achieves success through an incredible number of baby steps. The fact that we are not yet able to build a being like Data in Star Trek does not mean we cannot create some very usable applications in our professional lives. Baby steps have already been taken in the form of technological advancements, including optical character recognition (OCR), speech recognition, digital encoding of pictures and data, and smaller, faster computers. All of these are helping to pave the way for AI-based applications.

I work for a company that creates software for state and city legislatures. Our group is constantly under the gun to produce new and innovative software. To be competitive with the other vendors, we have to show that our product is better. Most important, we have to deliver the product on time.

Recently, we were up against a tight deadline and the pressure was starting to mount. People were getting testy and irritable, and everyone seemed to be waiting on the next guy to do their part. One of my colleagues jokingly informed the other developers that we would soon be installing the newly released Microsoft telepathy product. This could be used to read the minds of all the other developers and make sure everyone was on the same page.

Now, of course, there is no Microsoft telepathy software. Even with the phenomenal progress that has taken place in neuroscience, we still do not fully understand how the brain works, much less how to read one another's minds. Still, I can envision a day, not too far off, when we will have software that can anticipate what users are thinking and act on those predictions.

This book is not about reading minds. It is about utilizing a few artificial intelligence technologies to enhance the value of existing and new applications.

The four AI-related areas I cover in this book are not meant to represent everything the field of artificial intelligence has to offer. In fact, they are just a sampling of the somewhat unexplored potential for business-related applications. They are

- Agents
- Data mining
- Rule-based systems
- Speech processing

Who Is This Book For?

I wrote this book for developers who are not satisfied with "business as usual." For developers who are always looking to improve their code and their skill sets. For developers who surf development Web sites and read technical journals, hoping to make the applications they work on better.

This book is for any developer, contract or in-house, who is building enterprise-wide applications. The applications could range from those running standalone within the company to those dispersed geographically as Web applications. The companies could be of any size, large or small, but large organizations will probably find the most benefits.

Readers already familiar with Microsoft products, and especially with Visual Studio .NET, will get the most out of this book. It focuses on using AI-based techniques with Microsoft products. Readers already familiar with Microsoft SQL Server will be the most comfortable, since all of the applications utilize Transact-SQL (T-SQL) inside stored procedures.

What You Should Know

This book is not an introduction to programming with Visual Studio .NET. Many of the applications featured in the book are Web-based. It is assumed that the reader is already familiar with designing Web-based applications using ASP.NET. Readers should be familiar with such concepts as setting up a virtual directory and executing a Web application on their localhost machine. They should also understand the difference between server-side script and client-side script.

The sample applications are not intentionally complex, but they access code embedded in SQL Stored Procedures. These stored procedures contain some of the application logic in the form of T-SQL statements. The reader should have a basic understanding of T-SQL and know how to view stored procedure code using Microsoft's Enterprise Manager or Query Analyzer.

To simplify common operations such as data access and exception handling, the sample applications utilize Microsoft Application Blocks. These blocks contain code that demonstrates best practices for accessing common functionality. No additional installation steps are necessary for the reader to execute the sample code. However, readers interested in learning more about the functionality available in these blocks should visit the Microsoft Patterns and Practices Web site at http://www.microsoft.com/resources/practices/code.mspx.

A Note about Security

The sample applications in this book were designed to demonstrate key techniques presented in each chapter. **Although limited forms of security were considered, readers should not assume that the sample applications are production-ready.**

Readers interested in using the sample applications as a starting point for building their own solutions should thoroughly evaluate the code for security weaknesses. The Microsoft MSDN Web site (http://msdn.microsoft.com) contains many helpful resources for writing secure code. One resource readers may want to refer to first is a MSDN Webcast titled "Writing Secure Code—Best Practices." The Webcast, which originally played on October 12, 2004, is available through the Microsoft On-Demand Webcasts Web site at http://www.microsoft.com/seminar/events/webcasts/ondemand.mspx.

Why I Wrote This Book

I am the type of developer who is always pushing the envelope—looking for more. I eagerly await the emergence of new products and technologies, and I am always trying to make the next application I write better than the last one. That being said, I also have a limited amount of time every day to get everything done.

I wrote this book to explore and introduce some interesting technologies and products that I think are either unknown or unexplored. They all revolve around AI-based concepts that have been around for several years. This book seeks to introduce concepts that may have been viewed as too hard in the past.

The world is changing fast, and the development world even faster. Professional software developers do not have time to learn the foundation of every AI-related technology. We barely have enough time to wade through the documentation that accompanies a beta product. What we need is someone to give us an intuitive and standard way for accessing these technologies. We know that AI technologies can be useful, but we need a good way to get to them.

Sample Applications

This book introduces some AI-based concepts using Visual Studio .NET. Starting with Chapter 3, the book will present six practical applications that can be designed with Visual Studio.NET. All the code for the sample

applications is available in both Visual Basic and C# versions on the book's Web site at http:// www.awprofessional.com/title/0321246268 and http://www.cutsolutions.net.

The first two applications deal with the newly released Microsoft Speech Server product. This product allows you to create Web-based applications that communicate with the user through speech. Speech processing is already important to companies that want to automate their call centers. As society becomes more mobile, speech processing will be critical to enabling mobile workers. Readers interested in learning about this natural interface will want to focus on Chapters 2, 3, and 4.

Starting with Chapter 5, the book switches to utilizing Microsoft Analysis Services. Analysis Services can be used to build data-mining models against historical data in a relational database. These models can then be used to make predictions about the data. Data mining involves more than just querying a database or creating a fancy report for business executives. It involves uncovering trends and patterns in large quantities of data. In many cases, the trends and patterns discovered were not even thought to have existed.

Chapter 7 examines the concept of rule-based programming. Rules encapsulate the business logic of a program and are stored in a central repository. Each rule is assigned a priority, and the series of rules that is applied will depend on the program's input. Programs using this technique can be more reactive to their environments.

Software agents are explored in Chapter 8. Just as a sports agent represents the best interests of an athlete, a software agent is meant to represent the best interests of the user. An agent is generally independent, reactive, personalized, and communicative. Agents that are able to learn from their users are the most valuable and are thought to possess intelligent capabilities.

Why Just Microsoft Technologies?

This book deals exclusively with Microsoft technologies. The primary reason for this is that Microsoft technologies focus on the needs of businesses and therefore are in line with revenue-generating goals. Many developers today are already familiar with Microsoft tools because they serve as the building blocks of some key business applications. This instantly lowers the learning curve needed for exploring AI techniques utilizing them.

Microsoft has dedicated substantial funds to a vast research division, Microsoft Research, that reviews many current AI areas. The division has made many advances toward speech recognition and natural language processing, as will be explored in the first and last chapters.

Microsoft is generous in providing developers with knowledge. The numerous Software Development Kits (SDKs) available for free download from its Web sites are a testament to this. These kits significantly lower the learning curve associated with adopting new technologies. Since the primary goal of this book is to lower the learning curve associated with AI, it makes sense to utilize a few SDK's in the process.

This book is the start of what I hope will be a trend toward making AI-related technologies more accessible and easy to use. I hope to see more and more developers finding creative and insightful ways to harness these technologies. Have fun!