If you've tried to implement software agents in the past few years, you've probably heard about the Java Agent Development Framework. JADE's success arises from the fact that programmers can implement software agents according to broadly accepted Foundation for Intelligent Physical Agents specifications without having to study the specifications in great detail. Fabio Bellifemine, a JADE project leader who was also actively engaged in the FIPA standardization efforts, provides an interesting parallel history of FIPA and JADE in chapter 2 of Developing Multi-Agent Systems with JADE. However, it would be inappropriate to simply call this book a how-to guide on implementing JADE-based agents because it also describes the framework's low-level architecture and explains the motivation for its construction. Additionally, the book discusses related tools and add-ons that are useful for enhancing software agents' semantic features, launching them on mobile devices, and automating construction, deployment, integration, and debugging tasks.

To illustrate their concepts, Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood use a book-trading example (certain agents selling and buying books), which evolves as they describe new features and as readers move through the text. The authors present simple problems and their solutions to help readers become familiar with JADE's basic features before moving on to complex cases. For example, you can implement and launch your own selling and buying agents, which can discover themselves in a distributed environment and communicate their book-trading proposals. Chapter 5 answers some of the more advanced questions that several of our students who used the text in class asked—particularly, when and how to use semantically richer communication (ontologies and content languages) and complex interaction protocols.

However, the section about building ontologies is sometimes unclear because the authors don't explain all the keywords they use (the schema term, for example). Of course, theory is still supported through the book-trading example: you'll learn how to port your book-trading agents to mobile devices (chapter 8), deploy them automatically in a distributed environment (chapter 11), or enhance them with a semantically richer knowledge base and communication (chapter 12). The trading example's complete source code is available from the JADE Web site (http://jade.tilab.com), but only in its final version. Therefore, you have no access to the code developed in intermediary steps.

Chapter 8 describes launching JADE-based agents on mobile devices such as PDAs or mobile phones. This is also where you'll learn to develop simple applications using the Lightweight Extensible Agent Platform add-on. The LEAP project contributed significantly to porting JADE to Java 2 Micro Edition and wireless network environments. The authors not only present how to port book-trading agents to the JADE-LEAP platform here but also the mechanisms behind this approach. In addition, this chapter introduces mobile technologies and compares their limitations and differences with traditional hardwired networks and desktop devices, which makes it easier to understand the specificity of
solutions presented thereafter. In particular, the presentation of the LEAP project is very clear and leaves you with a firm understanding of the differences between the LEAP and JADE frameworks. In contrast to the PDF tutorial attached to the JADE documentation, the chapter comments on and explains questions related to particular steps in JADE-LEAP application development.

The authors make strong distinctions between agent-based design (abstraction and methodology) and agent technologies (using agents in a particular design’s implementation) to describe the JADE framework’s features in a broader technological context. They compare agent mobility (chapter 6) to other paradigms such as client-server architecture and the remote execution approach. Their description of agent-based abstraction, however, failed to convince us that JADE (and software agents in general) could solve certain problems. As an example, the authors present the architecture of two solutions—JADE Semantic Framework and Jadex (http://vsis-www.informatik.uni-hamburg.de/projects/jadex)—that turn JADE agents into Belief Desire Intension-oriented agents. The first solution is based on a BDI abstraction proposed by M. David Sadek; the latter is based on one presented by Anand S. Rao and Michael P. Georgeff. However, you can find all the presented information on each project’s Web site, so the book would be much more valuable if the authors compared the different aspects of BDI abstractions and architectures with the JADE Semantic Framework and Jadex. The authors’ book-trading example is relevant for agent abstraction because sellers and buyers appear to be independent entities, but explaining why traditional approaches would be a poor choice for this example would better benefit readers and users new to JADE—as evidenced by the posts on the JADE mailing lists in which new users appear to have a poor understanding of related approaches and technologies.

The interesting raisin in the cake is chapter 7, which describes the JADE internal architecture. It clearly describes how JADE lets potential developers enhance it with their own useful kernel services.

As a guide, this book is much better than online documentation because it’s more comprehensive. However, it still suffers from a lack of comparisons and discussion as to why the JADE framework would fit into particular problems. Therefore, we would recommend the book to those who have already read the documentation provided on the JADE Web site.

Maciej Gawinecki is a senior researcher at the Systems Research Institute, Polish Academy of Sciences and a PhD student at the International Doctorate School in Information and Communication Technologies in Modena, Italy. Contact him at maciej.gawinecki@ibspan.waw.pl.

Grzegorz Frąckowiak is a researcher at the Systems Research Institute, Polish Academy of Sciences.

Cite this article: