Vision without implementation is hallucination.

–Benjamin Franklin (1706–1790)

In addition to usability and the selection of useful adaptations, Höök (2000) identifies development methods as a crucial prerequisite for a widespread adoption of adaptive interfaces. We presented an approach for user modeling from basic events in Chapter 3 and a set of adaptation patterns for interactive systems in Chapter 4. On this basis, we devised an adaptation framework as part of this book to allow a developer to integrate the presented approaches into interactive systems more easily and conveniently. This generic adaptation framework provides a reusable platform for adaptive interactive system and improves the task of developing these systems. For this purpose, it integrates user modeling and adaptations. First, using an existing framework reduces the development time, because the developer does not have to implement all the components of the framework. In addition, this framework comprises a number of user modeling algorithms and adaptations, which thus do not have to be implemented. Second, the developer may avoid common pitfalls of adaptation architectures by employing an existing architecture rather than developing a new framework. In addition to supporting developers in creating adaptive interfaces, the framework also serves as a test bed for an evaluation
of the adaptation approach. In doing so, the framework shows the feasibility of our adaptation approach.

In this chapter, we present an adaptation framework for adaptive interactive systems. The main components of the framework are a user modeling component, an adaptation component, and a semantic layer. The user modeling component observes the user-system interaction and creates an abstract representation of the user. The adaptation component applies adaptations to the interactive system. For this purpose, it employs information stored in a semantic layer that serves as an abstraction of the interactive system. We created a reference implementation of this framework as an extension to a model-based development tool. The implementation includes the user modeling algorithms presented in Chapter 3 and the adaptations from Chapter 4. The reference implementation also serves as a test bed for an evaluation and verification of these approaches. We show the feasibility of the framework by implementing adaptive features in different interactive systems and performing user tests with these interactive systems. These systems include a digital TV system and an automotive dashboard system. Chapter 6 presents an evaluation of the user modeling algorithms and the adaptive test systems, which employ our adaptation framework.

This chapter is structured as follows. After an overview of the architecture of the framework, a semantic layer that serves as an abstraction of the interactive system is introduced. Thereafter, a user modeling component is presented that extracts information from the user-system interaction and computes further derivations with user modeling algorithms. Finally, the integration of adaptation patterns into the adaptation framework is discussed and a reference implementation of the framework is presented.

5.1 Overview of the Architecture

In the following, we present an adaptation framework called AdaGUIDE that integrates the concepts presented in previous chapters. The aim of the framework is to provide a reusable platform for adaptive interactive systems and to create a test bed for an evaluation and assessment of the adaptation approach we present in this work. Figure 5.1 gives an overview of the framework. A semantic layer, a user modeling component, and an adaptation component constitute the main components. The framework extends an interactive system. Triggered by user behavior, the user modeling component uses algorithms, such as the ones presented in Chapter 3, to model the user. An adaptation component integrates the adaptation patterns from Chapter 4 to adapt the interactive system.

A semantic layer creates an abstraction of the interactive system and other aspects relevant for adaptations. It serves two purposes. First, this layer creates a uniform representation of different components, such as the interactive