Chapter XII

Augmenting UML to Support the Design and Evolution of User Interfaces

Chris Scogings, Massey University, New Zealand

Chris Phillips, Massey University, New Zealand

Abstract

The primary focus in UML has been on support for the design and implementation of the software comprising the underlying system. Very little support is provided for the design or evolution of the user interface. This chapter commences with a brief review of UML and its support for user interface modeling. Lean Cuisine+, a notation capable of modeling both dialogue structure and high-level user tasks, is described. It is shown through a case study that Lean Cuisine+ can be used to augment UML and provide the user interface support that is currently lacking.

Introduction

Considerable effort has been devoted to the development of models and tools to support the analysis and design of software systems, culminating in the Unified Modeling Language (UML) (Rumbaugh, Jacobson, & Booch, 1999). The primary focus has been on support for the design and implementation of the software comprising the underlying
system, referred to in (Collins, 1995) as the *internal system*. Very little support is provided for the design of the *external system* or user interface.

The user interface is a vital part of any software application. To the user, the interface *is* the system (Collins, 1995). It is important, therefore, to devote time and effort to the user interface when building and upgrading software. Indeed, research indicates that applications are often upgraded precisely because a new user interface is desired (Church & te Braake, 2002). The user interface may be redesigned to improve its usability, to add new functionality, or to take advantage of new technologies and visualisation techniques – for example, a system may be changed from a forms-based interface to a graphical user interface (GUI), or a help system may be added.

A user interface can be described at three levels:

- the visible interface (the ‘look and feel’);
- the underlying structure and behaviour, known as the *dialogue*, which describes the response of the system to user and system generated events;
- the high-level user *tasks* (sequences of actions) that must be supported by the interface.

These three views of the user interface need to be identified and separated so that during software evolution the visible interface can be changed independently of dialogue structure and user tasks — although these also can be modified or extended if required. The dialogue and task information is preserved through the use of appropriate conceptual models.

This chapter commences with a brief review of UML and its support for user interface modeling. Lean Cuisine+, a notation capable of modeling both dialogue structure and high-level user tasks, is then introduced. It is shown through a case study that Lean Cuisine+ can be used to augment UML by providing the user interface support that is currently lacking. The work is placed in context.

### UML and the User Interface

Prior to the introduction of UML, it had already been established that there was a lack of integration between the software engineering and HCI communities (Jacquot & Quesnot, 1997). Janssen, Weisbecker and Ziegler (1993) make the point that user interface tools cannot make use of the models developed with general software engineering methods and tools. Kemp and Phillips (1998) show that support for user interface design is weak within object-oriented software engineering methods.

As UML is primarily a collection of previously defined notations, no significant change has occurred. The approach taken in Eriksson and Penker (1998) is typical in that they state that the design of the user interface should be carried out separately but in parallel