Specifying the Apple Macintosh™ Toolbox
Event Manager

C. T. Burton, S. J. Cook, S. Gikas, J. R. Rowson
and S. T. Sommerville
Department of Computer Science, Queen Mary College, University of London,
327 Mile End Road, London E1 4NS, United Kingdom

Keywords: Formal algebraic specification; Larch; User interface; Apple
Macintosh™ Toolbox

Abstract: The Macintosh™ Toolbox is a collection of resources, stored in ROM,
of use to software developers building user interface components. This paper
documents an attempt to specify one part of the Toolbox: namely the data types
and routines for event-handling. The chief objective of the paper is to illustrate
an approach to building a formal specification in which, starting from simple
structures, the final specification is arrived at by a process of successive
modifications, generalisations, instantiations and translations. The paper also
constitutes an example of the Larch algebraic style, as well as the two-tiered
approach characteristic of the Larch family of specification languages.

1. Introduction

It is a common claim that, in the context of stand-alone micros, the Apple
Macintosh™ represents a paradigm of user-oriented design. Application builders
are exhorted to preserve the consistency and uniformity of the Macintosh user
interface both through large volumes of documentation (see Section 2.2, below)
and through provision of a user interface building environment, the User Interface
Toolbox. We present here our experiences in writing a formal specification, in
Larch, for a fragment of the Toolbox.

† This work has been undertaken as part of the Alvey HI 059 project "Formal Aspects of Interactive
Dialogues", supported under SERC Grant No. GR/D/1646.8.
Correspondence and offprint requests to: C. T. Burton, Department of Computer Science, Queen Mary
College, University of London, 327 Mile End Road, London E1 4NS.
This paper is taken from the work of the "Formal Aspects of Interactive Dialogues" project at Queen Mary College, which has been concerned with putting formal methods to use in the area of user interface software design and engineering. The slogan that has emerged is that, in order to improve the design and behaviour of user interface software, there is a need for exact and perspicuous notations, within which designers and developers can express and reason about the structure and behaviour of software components.

We decided to consider from this point of view the "guidelines" for Macintosh applications developers, which are intended to ensure the consistency and uniformity of applications with the Macintosh User Interface Standard. In considering these guidelines, however, we found ourselves in need of a formal specification of at least part of the Toolbox in terms of which applications are written. This was the origin of our work on the Toolbox, which involved both the planning of an eventual overall specification and also the detailed investigation of a few reasonably self-contained components. One of these was the component known in Macintosh documentation as the Toolbox Event Manager (TEM). TEM is central to the role of the Toolbox in mediating between the underlying (largely invisible) Macintosh operating system, the applications software and the user's input actions. A formal specification of TEM is the core of this paper.

In what follows, we first describe TEM informally (Section 2) and discuss some general issues that arise in contemplating specifying it (Section 3). We then present in Section 4 a series of steps leading to a specification which is still somewhat removed from TEM in the behaviour that it specifies, and also from the language Larch in the specification formalism that it uses. Sections 5 and 6 bridge these two gaps, yielding a specification in Larch of TEM.

2. The Macintosh Toolbox

A great deal of thought and effort has been put into producing a user interface for the Macintosh which, according to Apple, "feels natural and comfortable to people who aren't computer experts as well as those who are" [Che85, p. 2]. To assist application programmers in achieving this, Apple produced the Toolbox, which is built into every Macintosh in ROM. It consists essentially of a large collection of machine-code routines, including those for creating and manipulating windows, pull-down menus, scroll bars, dialogue boxes etc. It can be thought of as occupying a level above the operating system, upon which it calls for lower-level operations. The Toolbox removes any need for the normal application programmer to call upon the operating system directly. The Toolbox routines fall into groups which in the documentation are called "managers" of the user interface features they support. Figure 1 reflects one kind of inter-relationship between these managers: higher level ones making calls upon lower-level ones (e.g. for graphics, fonts and desk accessories), via TEM.

2.1. The Event Manager

One of the principles of the Macintosh philosophy is that people should tell the computer what to do and not the other way around. The user controls the program's behaviour by clicking the mouse or typing on the keyboard; each such action constitutes an event for the program to respond to. TEM is thus the part