Chapter 3

Multimedia System Design

Design is really quite a complex issue. It covers the visual images and movies, the sound and music, the human-computer interface, the purpose of the product and of course its packaging and marketing.

At the most basic level, there is a question of how to organize the material and get it written to a CD-ROM disk. To do this, you will normally use a commercial software package known as an Authoring System.

3.1 Authoring Systems

An authoring system is a program which has built-in elements to help you create a multimedia presentation. Commercial systems have pre-programmed elements for the development of interactive multimedia, saving you time compared with raw programming. For example, such systems allow you to indicate ‘clickable’ areas of the screen display by drawing polygons around them, rather than specifying coordinates.

Authoring is arguably just speeded-up programming, with the authoring system being what allows you to go quickly. Even so, some knowledge of heuristics and algorithm design is usually needed.

There are several different models which an authoring system can use:

- Scripting language;
- Score;
- Card;
- Visual Programming.

3.1.1 The Scripting model

The Scripting model is the authoring method closest in form to traditional programming. The model is that of a programming language, which specifies multimedia elements (by filename),
sequencing, clickable hotspots, synchronization, etc. An object-oriented scripting language is usually offered; in-program editing of elements (still graphics, video, audio, etc.) tends to be minimal or non-existent.

The scripting model tends to be longer in development time (it takes longer to code an individual interaction), but generally more powerful interactivity is possible. Since most Scripting languages are interpreted, rather than compiled, the run-time speed gains over other authoring methods are minimal.

An example of this is Lingo.

### 3.1.2 The Score model

The Cast/Score/Scripting model gives a presentation which loosely resembles a music score. The various elements are shown on horizontal “tracks” with simultaneity shown via the vertical columns.

The most popular software to use this model is Director, which is used in the creation of many commercial applications. These programs are best suited for animation-intensive or synchronized media applications; they are easily extensible to handle other functions (such as hypertext). Director also includes a scripting language, Lingo, so is really a hybrid system.

### 3.1.3 The Card model

The Card model uses the index card as a model: viewers see the current ‘card’ but there are hotspots (clickable regions) which either cause something to happen (a short animation of the object clicked, for example) or take you to another card (like a web-like hyperlink).

It provides a great deal of power via the incorporated scripting language, but suffers from the index-card structure. It is excellently suited for Hypertext applications, and very well suited for navigation-intensive applications, such as Cyan’s original “Myst” game (made with Apple’s HyperCard).

Such programs are also easily extensible and so they are widely used for shareware applications. The best applications allow all objects (including individual graphic elements) to be scripted. It is also a low-cost prototyping method, prior to compiled-language coding.

### 3.1.4 The Visual Programming model

This tends to be the speediest (in development time) authoring style; it is well suited to rapid prototyping and short-development time projects. The core of the model is the Icon Library, containing the possible functions/interactions of a program. This is called visual programming because you ‘program’ by interactively drawing lines between icons, in order to create the data path between them.

These programs tend to be the slowest at runtime, because each interaction has to be interpreted on the fly.

### 3.2 Other specific tools

In principle, you do everything in the authoring system. In practice you may from time to time need to use more basic tools, such as:

- Paint programs for still images (photos, original digital artwork)
- Illustration (draw) programs for still images
- Modeller/renderer systems for 3D images
- Video digitizing/editing programs
- Video effects programs
- Audio sampling/editing programs
- Word processors (script and program text)
- Database programs
- Animation programs

What this really shows is that preparing multimedia is complicated. You can no more do it with one piece of software than you can assemble a magazine with only editorial skills: somebody has to provide the artwork, the photographs and the text. Indeed for multimedia you also need the sound, the movies and the interlinking structure.