5

LIVE update: archaeological courseware using interactive video

Clive Ruggles
(School of Archaeological Studies, and, Department of Computing Studies, University of Leicester, Leicester LE1 7RH)
Jeremy Huggett
(Department of Archaeology, The University, Glasgow G12 8QQ)
Steven Hayles
Howard Pringle
Ian Lauder
(Department of Pathology, University of Leicester, Leicester LE2 7LX)

5.1 Introduction

Interactive video (IV) technology, whether analogue- or digital-based, has huge potential in a wide variety of teaching and training applications where the skills of interpreting visual material are fundamentally important, as in the teaching of archaeology. It is ideal for independent student use or, with networked workstations sharing a single resource, for tutorial group teaching.

The initial phase of the LIVE (Leicester Interactive Video in Education) project was sponsored by the UGC/Computer Board under their Computers in Teaching Initiative (CTI). Its aim was to produce a system using interactive video (IV) technology to teach archaeology to undergraduates. A laser videodisc ('The Archaeology Disc') was produced in 1988, together with basic authoring facilities enabling tutorials to be written for the RM Nimbus PC1 computer linked to a Philips videodisc player (Ruggles 1988).

The LIVE project was not the only project at Leicester University during the late 1980s concerned with the development of teaching and training applications using interactive video. A collaboration between nine UK University Pathology Departments (Mercer *et al* 1988) produced two videodiscs for teaching human and vetinary pathology ('UK PATH 1' in 1987 and 'UK PATH 2' in 1989) together with the 'VIPA' authoring system which has enabled several hundred tutorials to be written for the BBC Master computer linked to a Philips videodisc player (CTICM 1989).

Although there were considerable pedagogical differences of approach between two such diverse subject areas, interaction between the two projects turned to collaboration through concerns about portability and future-proofing for IV courseware. There was considerable concern within the Pathology project for the maintenance of existing VIPA tutorials: there was an increasingly pressing need to transfer them to different and more modern hardware so that they could be used there and improved to make use of updated technology. Similarly, progress towards portability in LIVE was impeded by the lack of off-the-shelf software with the required low-level functionality that might enable high-level software to be built upon standard kernels for video handling and merging, graphics and windowing.

The result of the collaboration is the current phase of LIVE, whose crucial concern is for the longevity of IV courseware, which implies ease of maintenance and portability between authoring systems. The philosophy of the project is that these aims can be achieved through standardisation. Hence the project is concerned with the standardisation of concepts for the development and execution of courseware using multi-media technology in general and IV technology in particular. 'Academic portability', that is, applicability to a wide variety of subject areas where teaching methodologies may be very different, is an important concern. Pathology is a training area where a rigid goal-oriented approach to computer-aided learning (CAL) prevails: archaeology is an area where open-ended exploration is generally much more valuable. The combination of these subject areas, together with others (notably geography) that are also involved in the current phase of LIVE, has helped considerably in preventing too discipline-specific an approach. The current phase of LIVE is sponsored by IBM UK Ltd., the Leverhulme Trust and the CEC COMETT initiative.

5.2 Standardisation: issues and goals

5.2.1 The importance of standardisation

Interactive video is an important component of fast-developing multi-media technology. Interactive laser videodiscs, on which large numbers of images are stored in analogue form with fast access times, have been available for some years. Since the development of CD-ROM drives in 1984, allowing personal computers to access over 600 megabytes of prerecorded digital information, developments in interactive digital video have been rapid. Recent technological developments include CD-I (compact disk-interactive), a standard for the storage and retrieval of multi-media information on compact discs, and DVI (digital video interactive), an alternative to CD-I allowing up to 72 minutes of full motion video together with FM-quality audio to be stored on compact disc and decompressed in real time (Fox 1989). The appearance of erasable optical discs in 1988 points the way for the storage, manipulation and retrieval of large amounts of multi-media data by individual PC users.

Progress in off-the-shelf software, and particularly in authoring systems, is equally rapid. Each such system, however, is dependent upon a specific range of hardware and software, such as Apple Macintosh-based multi-media systems running under HyperCard. While the possibility of transferring material between such systems is likely to increase, through the development by manufacturers themselves of standard interfaces and standard text-file transfer formats, this facility is a low-level one, equivalent perhaps to the transfer of text and basic formatting instructions between different word-processing systems. There is a distinct danger that high-level structural inconsistencies between objects (such as CAL tutorials) created on two different systems, arising because each system has a fundamentally different abstract model of the objects in question, will render impossible any largely automatic interconversion.

For this reason it is crucial that we standardise concepts for the development and execution of courseware using multi-media technology in general and IV technology in particular.

5.2.2 Standardisation: the LIVE approach

The emphasis of the project is upon abstract structuring for portability, which should be independent of hardware considerations such as whether the images in question are stored (for example) in analogue form on laser videodisc or digitally on CD-ROM.

Extensive dialogue has taken place during the last two years with the developers of CAL tools in a variety of disciplines, and in particular with Pat Harkin and Mike Whittlestone, the authors of VIPA. The aim of these discussions was to make full use of existing experience in developing an abstract model for the structuring of CAL courseware using interactive video alongside more conventional text and graphics capabilities. Some major challenges confront such a development. First, the model must be conceived on a sufficiently high level of abstraction that is is not tied to a particular hardware configuration. Secondly, it must be easily extendible (e.g. to other media). Finally, and perhaps most critically, it must be 'academically portable', that is it must take account of, and attempt to encompass, the range of fundamentally different teaching methologies which pertain in different subject areas.

The structure model that has been developed (Ruggles 1989c) is expressed in VDM-SL, the Specification Language of the Vienna Development Method (Jones 1986). This is one of several formal (i.e. mathematically based) notations for software specification whose syntax and semantics are precisely defined. Their use generally leads to clarity, conciseness and the removal of ambiguities in specifications. They are useful both in the development and expression of data structure models and, where (as in VDM) the formal specification language is accompanied by a 'development method', provide a basis for the verification that software conforms to its specification. Formal methods have become increasingly popular as questions such as software reliability, conformance and standardisation become ever more important, and are set to become as much an everyday part of software engineering as are standard techniques of applied mathematics within long-established branches of engineering (Ruggles 1990b).

In the development of the LIVE data structure models we have used a modular version of VDM that is in the process of standardisation by the International Organisation for Standardisation (Andrews 1988). This modularity facilitates the description of large systems with complex data structures and allows us to follow an object-oriented style in which each module encapulates a class of data objects at a certain level of abstraction.

Alongside the structure model a standard text file ('metafile') format has been developed (Ruggles 1990a) for expressing particular instantiations of the structure, i.e. particular tutorials. The metafile is the vehicle for transferring courseware between different authoring and display systems. Our aim is that portability and future-proofing will be achieved by courseware authoring systems (i) using the standard structure model for modelling courseware and (ii) incorporating facilities to read and write standard metafiles. The metafile definition is expressed in EBNF notation.

A courseware development system is currently being developed at Leicester. It incorporates the tutorial struture model referred to above and also incorporates facilities to read and write metafiles. While it is hoped that the system will be portable between a number of hardware configurations, the real value of the LIVE approach depends upon propagating the structure model and metafile format so that they are used by the authors of other multi-media CAL authoring systems.

5.3 The LIVE structure model

5.3.1 General description

Central to the structure model is the concept of a 'tutorial' which provides a student with particular material within a structured, interactive learning environment. The degree of constraint upon the student can be determined by the tutor, and will be influenced by the prevalent teaching methodology in the subject area in question. One extreme constitutes a rigid computer-aided learning (CAL) facility where a student follows paths set by the tutor, the exact route possibly depending upon the student's response to certain questions. This is the philosophy behind most existing VIPA tutorials.

The other extreme allows unrestricted free browse and response without impeding the student's progress in any way. This facility is provided by incorporating an 'image database' (Ruggles 1989a, Ruggles 1989b) alongside a conventional database, an image database being a generic data structure expressing the abstract relationships between the images on a videodisc or CD-ROM. An image database might, for example, contain structures allowing the user to explore a given object, obtaining different views by moving, turning, or zooming in and out.

Ratings may be assigned at various points within the tutorial structure which can be used as a factor in an assessment scheme.

5.3.2 Example extract from the formal definition

The following module which forms the formal definition is included here in order to give the flavour of the formal definition. Thus no further comments are added to those included within the definition itself.

```
module PresentationDefinition
```

A *Presentation* comprises *Presentation Objects* of various types: text presentations, picture presentations, image presentations, movie presentions *etc.* The metafile specifies how, ideally, these should be displayed — for example, an image and a text presentation together, followed by a text and two picture presentations together, and so on. Whether particular combinations of presention objects can in fact be displayed together is a function of the target hardware — for example, early merge cards for interactive video only allow a single full-screen image to be overlaid with a single text/graphics screen. How simultaneous display is to be achieved (e.g. through a windowing interface) or, if it is not possible, how a comprimise should be reached, is up to the application.

Various modes may be set by the tutor to determine how a student may traverse the presentation.

```
imports

    from LocalGraphicsDefinition:

         types
           Rectangle

    from IdentifiersDefinition:

         types
           PresentationObjectId

exports

    types

         Presentation

    operations

         ObjectInPresentation : Presentation → set of PresentationObjectId

definitions

    types

         Presentation :: pvm : PresentationViewMode
                          ptm : PresentationTraversalMode
                          pus : seq of PresentationUnit

         PresentationViewMode = {COMPULSORY, OPTIONAL}

         PresentationTraversalMode =
                                     {FORWARD ONLY,
                                     FORWARD ANDBACKWARD ONLY,
                                     DIRECTACCESS}
```

A *presentation* may be flagged by the tutor as compulsory or optional.

A *presentation* comprises a sequence of self-contained *presentation units*. The *presentation traversal mode*, again set by the tutor, determines whether these may only be run through forwards, or else forwards or backwards, or in any manner. Presentation units allow a presentation to be specified as a sequence of discrete parts, e.g. a text screen followed by a graphics screen followed by an image.

```
PresentationUnit    ::   ie  :  InitialEvent
                         ses :  SubsequentEvents
                         mm  :  MeddleMode

InitialEvent         =   ObjectPresentations

SubsequentEvents     =   seq of SubsequentEvent

SubsequentEvent     ::   aos :  AddedObjects
                         dos :  DeletedObjects

AddedObjects         =   ObjectPresentations

DeletedObjects       =   PresentationObjects

MeddleMode           =   set of MeddleOption

MeddleOption         =   {CANHIDEOBJECTS, CANMOVEOBJECTS,
                         CANRESIZEOBJECTS, CANDISTORTOBJECTS}
```

A *presentation unit* comprises a sequence of one or more *presentation events*, triggered in some way by the student, at each of which a number of presentation objects become available and/or cease to be available. Whether a student may meddle, e.g. temporarily move or remove at any time one or more available presentation objects in order better to see the remaining ones, is determined by the *meddle mode*. However, the sequence of events themselves may not be altered. A presentation unit which allows maximal meddling and consists of single event, gives the student full control over the display or removal of a number of presentation objects.

```
ObjectPresentations          =   seq of ObjectPresentation
ObjectPresentation           ::  poid :  PresentationObjectId
                                 pop  :  [PresentationObjectPosition]
PresentationObjectPosition   =   Rectangle
PresentationObjects          =   seq of PresentationObjectId
```

Where a presentation object forms part of a presentation unit, a rectangle within which it must be displayed may be specified (in suitable screen co-ordinates). If not, the application has the whole screen at its disposal. At an event, the screen position of an existing presentation object may be changed. The ordering of the objects at an event is taken to be a priority list specifying which objects should overlay others in the event of overlap, later objects in the list overlaying earlier ones. Objects added at later events overlay already-existing ones in the event of overlap.

```
operations

   ObjectsInPresentation (p : Presentation) r: set of PresentationObjectId

   pre    TRUE

   post   r= ⋃{pu ∈ p.pus} ( ⋃{op ∈ pu.ie} op.poid
                            ∪ ⋃{se ∈ pu.ses} ⋃{op ∈ se.aos} op.poid
                            )

end PresentationDefinition
```

5.4 The LIVE metafile definition

5.4.1 General description

Our metafile definition results in metafiles which are verbose but which (i) are wll suited to human interpretation and (ii) facilitate automatic parsing. Since one of our main objectives is to encourage developers of teaching systems to incorporate facilities to read and write our metafiles the latter point is considered extemely important. Point (i) is also imortant if, as we consider likely at least in the early stages, metafiles are edited directly by users in the absence of suitable authoring systems. (The Leicester system will not include full authoring facilities until a later stage). It would, however, be trivial to add a set of short alternatives to the long tokens currently specified.

5.4.2 Example extract from the metafile definition

The following extract from the formal definition is included here in order to give the flavour of the EBNF definition. Thus no further comments are added to those included within the definition itself.

Presentations

```
presentationDefinitions ::= 'BeginPresentations', presentationDefinition, {';',
        presentationDefinition}, 'EndPresentations';

presentationDefinition ::= presentationHeader, PresentationBodyDefinition;

presentationHeader ::= 'Presentation', presentationId, presentationTitle;
```

These definitions govern how a presentation is defined in the global list.

```
presentationBodyDefinition ::= 'BeginPresentation', [presentationViewModeDeclaration,
        ';'],[presentationTraversalModeDeclaration,';'],[meddleModeDeclaration,
        ';'], presentationUnitsDeclaration, 'EndPresentation';
```

If either the presentation view mode or traversal mode are not declared, then the global default is assumed. The meddle mode, if declared, is taken to be a default for all the units in the presentation.

```
presentationViewModeDeclaration ::= 'PresentationViewMode', presentationViewMode;

presentationViewMode ::= 'Compulsory' | 'Optional';

presentationTraversalModeDeclaration ::= 'PresentationTraversalMode',
        presentationTraversalMode;

presentationTraversalMode ::= 'Forward Only' | 'ForwardsAndBackwardsOnly' |
        'DirectAccess';
```

```
presentationUnitsDeclaration ::= presentationUnitDeclaration, {';',
        presentationUnitDeclaration};

presentationUnitDeclaration ::= 'BeginUnit', [meddleModeDeclaration], eventsDeclaration,
    'EndUnit'
```

If the meddle mode is not declared, then the default (that for the presentation, if declared; otherwise the global default) is assumed.

```
eventsDeclaration ::= initialEventDeclaration, {';', subsequentEventDeclaration};

initialEventDeclaration ::= 'BeginInitialEvent', objectPresentationDescriptionList,
    'EndInitialEvent';

subsequentEventDeclaration ::= 'BeginEvent', [addedObjectsDescription],
    [deletedObjectsDescription],'EndEvent';

addedObjectsDescription ::= 'BeginAddedObjects', objectPresentationDescriptionList,
    'EndAddedObjects';

deletedObjectsDescription ::= 'BeginObjects', presentationObjectdescriptionlist
    'EndDeletedObjects';

objectPresentationDescriptionList ::= objectPresentationDescription {';',
    objectPresentationDescription};

presentationObjectDescriptionList ::= presentationObjectDescription,{';',
    presentationObjectDescription};

presentationId ::= identifier;

presentationTitle ::= title;
```

The objects within a presentation may be referred to directly or indirectly. If an object has to be referred to more than once, e.g. to be displayed and then subsequently be deleted or redisplayed in a different position, then indirect referencing must be used.

5.4.3 Example extract from an actual metafile

The following is a short extract from an actual metafile definition.

```
 Presentation "Example presentation"
 BeginPresentation
   BeginUnit
     BeginInitialEvent
       PresentationObject "Accompanying text"
       BeginTextPresentation
         BeginPara
           "Here are a couple of pictures relating to Avebury."
         EndPara
       EndTextPresentation;
       PresentationObject "Picture no.1"
       BeginImagePresentation
         Image 08176;
         BeginCaption
           BeginPara
             "Avebury - flint knife, scraper"
           EndPara
         EndCaption
       EndImagePresentation;
       PresentationObject "Picture no. 2"
       BeginImagePresentation
         Image 08196;
         BeginCaption
           BeginPara
             "Avebury - Grooved ware pot"
           EndPara
         EndCaption
       EndImagePresentation
     EndInitialEvent
  EndUnit
EndPresentation
```

The metafile describes a 'presentation' consisting of two captioned pictures together with accompanying text, displayed together. No screen positions are explicitly specified, so by default it is up to an individual system to decide how to manage the display.

5.5 Prospects for archaeology teaching using the LIVE system

A new courseware development system is currently being developed at Leicester to run initially on networked IBM PS/2 machines running DOS4 and Microsoft Windows and making use of the facilities provided by new DVA4000 video merge cards supplied by VideoLogic Ltd. Modular design should ensure its eventual portability to the OS/2 operating system and Presentation Manager, to other PCs running DOS and Microsoft Windows, and to UNIX machines running X-windows. Facilities to read and interpret metafiles are being incorporated.

The LIVE system would handle the presentation defined in section 5.4.3 by displaying the two captioned pictures and the accompanying text in three separate windows, initially in default positions on the screen. The user (student) would then be free to manipulate the windows before moving on to the next part of the tutorial. The author (tutor) may restrict the degree of interaction available to the student if he or she so wishes.

Facilities to browse simple image databases were added in the autumn of 1990. This facility enables the system to make use of the many image structures, such as grids of maps and site walkabouts, incorporated on the Archaeology Disc (see Ruggles 1988).

During 1991, it is planned to add an authoring system, so that the entire LIVE Courseware Development System should be completed by the end of the year. In the meantime, metafiles must be created using conventional editing facilities and parsed upon input to LIVE (although a program has already been written that converts existing VIPA tutorials into metafiles).

It is planned to use tutorials running under the new LIVE system, together with the Archaeology Disc, in undergraduate courses at Leicester during 1990–91. Soft copy of the latest full metafile definition together with an example metafile will be available through the Archaeological Information Exchange following CAA90. The Leicester system, running DOS and MS-Windows, will be available at a nominal charge to Higher Education Institutions. The Archaeology Disc is available from Clive Ruggles at Leicester University.

Bibliography

ANDREWS, D. 1988. "Report from the BSI panel for the standardisation of VDM", *in* Bloomfield, R., Marshall, L. S., & Jones, R., (eds.), *VDM '88: VDM — the way ahead*, pp. 74–78. Springer-Verlag.

CTICM 1989. *Computers in Teaching Initiative Centre for Medicine—Update*, 1. Department of Pathology, University of Bristol.

FOX, E. A. 1989. "The coming revolution in interactive digital video", *Communications of the ACM*, 32 (794–801).

JONES, C. B. 1986. *Systematic Software development using VDM*. Prentice-Hall, London.

MERCER, J., J. H. PRINGLE, M. J. L. RAE, P. J. R. HARKIN, & I. LAUDER 1988. "How do we teach Pathology? — The laser videodisc and computer-assisted learning", *Journal of Pathology*, 156: 83–89.

RUGGLES, C. L. N. 1988. "Software for the Leicester Interactive Videodisc project", *in* Rahtz, S. P. Q., (ed.), *Computer and Quantitative Methods in Archaeology 1988*, International Series 446, pp. 523–542. British Archaeological Reports, Oxford.

RUGGLES, C. L. N. 1989a. "An abstract model for structuring of an indexed set of images, 1: A first approach", Technical Report 17, Department of Computing Studies, University of Leicester.

RUGGLES, C. L. N. 1989b. "An abstract model for the structuring of an indexed set of images, 2: Simple image structures", Technical Report 27, Department of Computing Studies, University of Leicester.

RUGGLES, C. L. N. 1989c. "An abstract model for the structuring of generic CAL courseware using interactive image storage and retrival, 1: A structure specification in VDM-SL", Technical Report 30, Department of Computing Studies, University of Leicester.

RUGGLES, C. L. N. 1990a. "An abstract model for the structuring of generic CAL courseware using interactive image storage and retrival, 2: A metafile specification in EBTF", Technical Report 34, Department of Computing Studies, University of Leicester.

RUGGLES, C. L. N., (ed.) 1990b. *Formal Methods in Standards*. Springer-Verlag, Berlin and London.