

Information Technology and Legal Education: Towards 2000

9th & 10th April 1992

Sponsored by

Context Ltd; Linklaters & Paines; Needham & James



Hypertext for Computer Assisted Learning in Law -The Use of "Soft Links"

Valerie Finch, Edward Finch

Napier Polytechnic, Edinburgh

Keywords: Computer assisted learning. Hypertext for CAL tutorials. Using 'soft links' programmed with LOGIX.

Abstract: The paper explores the potential for using a hypertext system (Guide) for the production of CAL tutorials comprising; simple explanations of basic legal principles, reading lists, Statute and case notes, and simple revision questions. As well as the facilities provided by Guide hypertext, programmed links are used so that use of the tutorial is possible without the user needing to use menus. Records of use are maintained by a signing on and off procedure. The programming uses the Logiix language that is part of the GUIDE system.

Introduction

Recent papers and conference demonstrations have shown that there are benefits in using Computer Assisted Learning in Legal education. However there is a major problem in a small jurisdiction such as Scotland; there are not, and it is unlikely in the immediate future that there will be, "ready made" learning packages. Consequently we have found it necessary to develop our own materials. A further problem is that the level of "computer literacy" amongst law lecturers in our institution is minimal.

In order to develop Computer Assisted Learning materials in our institution the following criteria were identified:-

- a) The teaching materials must be readily accessible to law students without specialist computing expertise.
- b) The system should be capable of being used by the students in its initial development stages and be suitable for expansion and elaboration to meet future needs.
- c) The production of teaching materials must be feasible for academic staff with minimal computer literacy. Academic staff prefer to produce their own teaching materials to use in their own way and

modify them periodically.

d) Commonality of materials should be achieved where possible to save the duplication of materials e.g. statute and case notes, reading lists, articles etc.

Hypertext seemed to be the a suitable medium for the achievement of our aims. The features of hypertext which are particularly attractive for our purpose are:

1 For the student

- a) The interactive nature of a hypertext document. (Students find its use stimulating)
- b) It permits students to progress at their own pace.

2 For the author

- a) There is no requirement to map out the entire network of nodes of information in advance. This is especially useful at the pilot stage as it permits us to start on a small scale and build onto the system in an iterative process.
- b) The same authoring system can be used for different types of teaching materials as hypertext links can operate in several ways. Basic competence in one authoring system enables the tutor to create animated text, questions and answers as well as statute and case notes.

Using the GUIDE hypertext system a prototype has been produced of part of an integrated tutorial package on Contract Law with an accompanying Statute and Case note database. Guide was chosen because:-

- It runs on PCs using the Windows 3 Graphical User Interface and can utilise the integrated nature of this environment including the "Cut and Paste" of material between applications and the ability, on suitable hardware, to have more than one application running concurrently.
- Guide supports a variety of types of link both within and between documents.
- The software supports a variable node size ie a node may be a single word or phrase or a large document.
- The development of hypertext documents using Guide is intuitive, with visible menu bars and pull down menus so very little instruction is required before simple teaching materials can be produced.
- It is possible to economically deliver the material using the GUIDE Reader system. This also ensures that the integrity of the system is maintained as it prevents the material being edited by accident.

Structure of the Contract Law hypertext tutorial

The system is made up of nodes of textual information linked in a number of ways

1. Sequential linking.

A link from the end of one screen of text to the beginning of another screenful of information as if they were pages of a book. This is done by means of a reference button i.e. a point on the screen which has been designated as a link to a reference point at the top of the next screen in the predetermined sequence. "Clicking" the mouse onto the reference button initiates the display of the next screen.

2. Linking by association.

This is best explained with an example. In the Contract tutorial a reference in the text to the title of an Act of Parliament, may be designated as a reference button. Activating this button retrieves and displays on the screen the Statute and Case notes document with associated text at the top of the

screen. In the same way, a reference in the tutorial material to a legal rule deriving from an individual section of the Act may be a reference button linked to a reference point at the appropriate section in the statute note.

Links by association may be made to different nodes in the same document or to nodes in separate documents.

3. "Backtracking" links

It is important that when a user is browsing in a hypertext document and uses links to other documents or other nodes in the same document that he should be able to return to the point he had reached in the original text, should he wish to do so. Failure to provide a means of return can leave the user "lost in Hyper space"

One method of backtracking is to have a reference button back to a predetermined location e.g. the first screenful of related text. If there is a reference button from a summary of legal rules to a section of an Act, a reciprocal reference button could be provided back to the summary of legal rules. This creates a very rigid structure and we found that it ceases to be viable when there is a node in the network to which there are links from several different sources. It would be necessary to have a labelled reference button back to each possible departure node. Backtracking is possible using the "navigate" menu in GUIDE, but our aim is for the students to learn law without having to learn to use the application software first. The system should be as intuitive as possible.

To this end a "backtrack", button has been created using a command button, i.e. a button which initiates a programmed sequence of actions. In the contract law tutorial document clicking on a backtrack button navigates back automatically to whichever reference button was used as the departure point to reach the current node. In effect it is locating a "bookmark" in the same way as if when reading a text book, the reader leaves a bookmark at the point he had reached whilst looking up a case in a law report. This differs from a "home" button which always returns the user to the top of the document he is using, from where he has to navigate again to the point he had reached.

This ability to have "soft" links generated by the student as they use the teaching material allows the hypertext system to be an extension of the way in which the student learns rather than every link being predetermined. It allows links to exist between several separate documents without causing an incomprehensible tangle of links. The LOGiiX programming for this link is as follows;-

```
#LOGiiX
function Main()
begin
    Backtrack();
    {backtrack to previous point, but the Guide object needs to have its backtrack
    attributes set correctly}
end
```

The coding for the backtrack button and its graphic were created once, it was then copied to wherever it was required. In effect it became a part of the toolkit used to build the system. Authors do not have to make their own tools but may have to specify what they want from "software toolsmiths", who can create them. The "object orientation" of GUIDE facilitates the reuse of buttons and code in this way.

4. Expansion links

These permit the "unfolding" of text "behind" a word or phrase displayed on the screen. Textual

information can therefore be displayed for perusal in an extremely condensed form. Words or phrases can be designated as expansion buttons. If the cursor is clicked onto the phrase which forms the expansion button, that word or phrase disappears, to be replaced by more detailed information. The text revealed by the expansion button can itself contain further expansions or reference buttons allowing a hierarchy of levels of information. Expansion buttons have also been used in the law of contract tutorial as a method of revealing "answers" behind multiple choice questions.

5. Links to notes and definitions.

When unfamiliar words or phrases appear in ordinary text, it is often necessary to break the continuity of study to seek clarification on the meaning of the word or phrase. By using a note button a link can be created to a definition which is written in a separate node. When a word has been designated as a note button, clicking of the cursor on the button causes the definition to appear in a window superimposed over the text beside the relevant word. This definition is displayed only whilst the cursor key is held down, allowing it to be read with only minimal interruption to the reading of the main text. The definition document also contains the program code for the command buttons. This document can be edited directly in the GUIDE authoring system.

Contents of the hypertext system

The pilot teaching package consists of brief explanatory notes on exclusion and limitation clauses in contracts, followed by some multiple choice questions. This is intended to be one of a series covering the basic principles of contract law at an introductory level, with scope for expanding into more levels of detail in the future. Any number of reference buttons may link to a single reference point so it is possible to have links from several teaching packages or from several places in one tutorial to a particular case, or Statute, or section of a statute.

The statute and case notes document is made up of very brief notes, (case notes contain less information than a headnote in a law report). The intention is that a more comprehensive level of information can be incorporated using expansion buttons, whilst retaining the brief notes for introduction and revision purposes. It is not our intention that this should replace the use of the LEXIS and CELEX databases, but that it should provide a source of information in a directly accessible form.

The Statute and Case notes document is, in its entirety, a single node in the network. Each statute or case note within it is also a node and each section of a statute may also be a separate node.

Maintaining the statute and case notes as a separate document will obviate the need for duplication in tutorial programmes. Currently this document has only reference points from other documents and backtrack buttons. In order to avoid "hypertext spaghetti" we do not intend to put any reference buttons that link within this document but we will add note buttons and expansion buttons revealing more information. The statute and case notes will as they expand be available to any of the hypertext documents built with this system.

Visual cues are used to enable the student to recognize the type of object displayed on the screen. Standard text and graphics and colour are used to distinguish reference buttons and expansion points. Also the cursor or pointer is programmed to change as it passes over different types of object to reinforce the function of the different objects.

Signing on and off and maintaining records of use.

Particularly whilst the tutorial program is being piloted we need to establish the extent to which it is

being used. The signing on and off functions were programmed by the use of command buttons and LOGiiX program code. They enable staff to:-

- a) record the name of the user.
- b) record the time during which the tutorial was used

in addition it ensures that the system is restored to its starting state with all expanded sections closed when the user leaves the document.

The prototype for the LOGiiX code was supplied by the publishers of GUIDE and was adapted for use with our pilot system.

Conclusions

By beginning with small scale, simple projects in order to familiarise themselves with the use of hypertext it should be possible for academic staff to develop hypertext learning materials in about the same number of hours as a traditional distance learning package on the same topic. Our experience is that hypertext software such as GUIDE give authors significant flexibility in the design of CAL materials.

The manual input of text can be avoided by incorporating text from other sources if it is available as computer text files. The text in the contract tutorial and statute and case notes was typed in manually consequently the notes at this stage are very brief.

Initial impressions of the use of the tutorial on exclusion Clauses in contracts have been encouraging. The tutorial has been available as an additional learning resource and has been used by students on introductory law courses for non specialists e.g. Business Studies and Commerce Degrees. The students have requested that more materials be made available to them in this form. We will continue to develop hypertext teaching materials with the aim of reaching a stage where the preparation of such materials is taken for granted in the same way as lecture notes, tutorial papers etc.

In addition it has shown other staff the potential of CAL material that is custom built and therefore complements the actual curricula of our courses.

Appendix 1 - The LOGiiX program code

There are a number of sample codes supplied with GUIDE, however effective use of LOGiiX requires familiarity with high level programming. Once a program has been created it can be reused in any document.

BACKTRACK

```
#LOGiiX

function Main()

    begin

        Backtrack();
```

```

        {backtrack to previous point, but the Guide object
        needs to have its backtrack attributes set correctly}

end

```

Each GUIDE object has distinct characteristics, one of which is the way it reacts to its use. To keep track, objects can be set to store their location in memory, when the backtrack() command is activated it gets the details of the location that was stored and navigates to it automatically.

LOG IN AND RECORD TIME

```

#LOGiiX

{Logs user in. Definitions named %on_open will execute on document open.}

{Original definition supplied by OWL in LOGuX examples}

global name;

global test;

{Global values of student name and name of this test are available to all code fu

function main()

begin

    test := " Contract Unit 7";

    SetDocCaption((), test);

{Window title is the name of this test i.e.Contract Unit 7 }

    name  "Unidentified User";

    String1 := " opened the" + test + " at";

    ret := "\n";

    name := Ask("John Smith", "Please enter your name for the log file");

    hText := Open("LOG.GUI", 0, 0);

{Open the logfile}

    hTop  GetFocus(hText);

    time := DateTime();

    SetSelection(hText, hTop, 65535, 65535);

    MessageBox(name+String1 +time+ret);

    {Put a string into a message box}

    InsertText(hText, name+String1 +time+ret);

```

```

    {Save text to the logfile called LOG.GUI}

    Close(hText, 1+256);

    {Close the logfile}

```

```
end
```

EXIT SAVE, AND FOLD TO TOP LEVEL

```

#LOGiiX

{Function to exit with save}

global test;

global name;

{Use global values of name and test}

function main()

begin

    e:=Answer(1 +32,name," Finish Tutorial?");

    {1 is ok /cancel, 32 is ? in the dialog box}

    if(e=1)

        then

            begin

                toplevel(0);

                {toplevel folds all expansions}

                CloseAll(1 +256);

                {Close and save file with no interaction}

            end;

    end

end

{If the user selects Cancel then nothing happens}

```

LOG OUT AND RECORD TIME

```

#LOGiiX

{Logs the user out. This definition is named %on_close. It is not linked to any b

```

```
{In effect this is invoked by the close function above}

{Original provided by OWL in LOGuX examples}

global name;

global test;

function main()

begin

    String1  finished the" + test + "at";

    ret:="/n";

    {Create a text string}

    hText := Open("LOG.GUI",0,0);

    hTop  GetFocus(hText);

    {Open the log file}

    time := DateTime();

    Message:=name + String 1 + time +ret;

    messagebox(Message);

    {Put a message on the screen to keep the user happy}

    SetSelection(hText,hTop,65535,65535);

    InsertText(hText,name+String1 +time+ret);

    Close(hText,1 +256);

    {put the users name and time into the logfile and close it}

end
```

NB This code works in the GUIDE system, in the current version of the GUIDE Reader software the logfile has to be a text file rather than a guide document. This limitation is not a restriction for this function.

Bibliography

Report of BILETA enquiry into the Provision of Information Technology in United Kingdom Law Schools. July 1991

Computer Assisted Instruction in Law;- A Primer, R.P.Jones 1987

As We May Think. V.Bush. In The Atlantic Monthly Vol 176 (from Hypertext and Legal Structure.

P. Wahlgren.)

A Hypertext Information Retrieval System for Lawyers. Eve Wilson 1989.

Hyperlaw, Desmond Painter 1989

The Hypertext as an effective information Retrieval Tool for the Final User, M. Agosti, G.Gradenigo P. Mattiello. 1990

Hypertext for Legal Documentation, R. M. Di Giorgi, B. Inghirami, M. Ragona, R.Mannucci, A. Archi. 1990

Storing , Accessing, and Communicating Information, Databases, Expert Systems and Hypertext. D. van der Merwe.

Authors

Valerie Finch and Edward Finch

Napier Polytechnic

Sighthill Court

Edinburgh