Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment

T.J. Brailsford[1]; C.D. Stewart[2]; M.R. Zakaria[1] & A. Moore[3]

[1]School of Computer Science and Information Technology, University of Nottingham, Nottingham, UK.
[2]School of Life and Environmental Science, University of Nottingham, Nottingham, UK.
[3]School of Civil Engineering, University of Nottingham, Nottingham, UK.

Abstract

The WWW is having increasingly profound effects upon education at all levels. When carefully implemented, web-based educational technologies can offer a cost-efficient means of maintaining teaching quality in the face of almost universally declining *per capita* resources. In this paper we will first examine the current situation with respect to technology-based learning, and highlight what we see as some of the key concerns with the current panoply. We will then go on to describe the major features and benefits of WHURLE, an experimental adaptive learning environment for the web. WHURLE is an XML-based system that stores educational content in atomistic constructs called chunks. Lessons are created, by educators, which provide a default pathway through the available chunks. This default narrative is adapted to the needs of individual learners by reference to their user profile. WHURLE also provides a robust linking system, and automatically generated navigation around the docuverse of the lesson.

By embodying a pedagogically sound design, we hope that WHURLE will address many of the deficiencies of most currently available integrated learning environments.

Keywords:

Integrated Learning Environment, Adaptive Hypertext, Linkbase, Autonavigation, XML, XSLT

Word Count:

approx. 6500

Introduction

The WWW is rapidly having profound effects - both qualitative and quantitative - upon education. If the new technologies are carefully implemented they potentially offer a cost efficient means of maintaining teaching quality in the face of almost universally declining *per capita* resources. They also offer unprecedented opportunities for remote teaching collaborations through the use of both distance and distributed models of

technology-mediated education (**Davies, 1999**, **Anderson & Jackson, 2000**).

Distance education uses an outreach model of online delivery, with centralised teachers and resources using the Internet to reach a dispersed population of students. The distributed model of education is quite distinct from this, in that students and teachers at one or more centres make use of distributed resources. One of the most exciting possibilities opened up by the new technologies is that of linking geographically separated centres of technology-mediated learning into a 'web' of distributed collaboration. If partners with complementary strengths and specialisations come together in this manner, both the cost-effectiveness and the overall efficacy of their teaching can be substantially increased. However, there are still both pragmatic and pedagogic barriers to the production of effective and efficient online learning resources.

In this paper we shall describe an experimental Integrated Learning Environment - WHURLE (Web-based Hierarchical Universal Reactive Learning Environment) [1] that we have developed (**Brailsford et al, 2001**; **Moore et al, 2001**). This is an ongoing project that contains a number of novel features.

The Need for Integrated Learning Environments

The most challenging problem is that of pedagogical design for learners now known to be highly diverse in respect of their pedagogical needs and preferences (**Valley, 1997**). Other challenges are almost as daunting. For example, the development costs of multimedia tend to be high. It is also the case that Technology Based Learning (TBL) products will rapidly depreciate if local institutions do not possess the technical and pedagogical skills (and the media resources) required for regular maintenance and updating. In addition to this extensive technical infrastructure is required for the effective and efficient delivery of TBL material. It is important that TBL development quickly ceases to be the domain of the "amateur enthusiast" - a situation that has existed for many years, and in too many institutions is still the norm rather than the exception. As a consequence, most TBL development still tends to be "custom development", with teaching products being implemented to fulfil specific needs, not to provide an enduring and transferable infrastructure for TBL delivery.

Generic software solutions known as Integrated Learning Environments (ILEs), have emerged in response to the problems caused by such inflexible TBL development. ILEs are usually designed to be discipline-independent, and thus hold out the prospect of greatly simplifying the process of implementation by partly automating the delivery of content and (to a greater or lesser extent) the burden of pedagogical design.

Approaches to Learning Environments

Ever since its early days, the WWW has been used as a vehicle for distance learning (**Dimitriyannis, 1994**; **Ibrahim, 1994**; **Speh et al, 1994**), and a number of web-based ILEs have been developed - of which the best know are WebCT [2] (**Goldberg et al, 1996**), TopClass [3] and Lotus Learning Space [4] . We shall here consider some of the key characteristics that any effective ILE should embody.

It is widely believed amongst the proponents of TBL that the learning experience can be substantially enhanced by multimedia, and there is both anecdotal and experimental evidence to support this (see the review by **Najjar,1996**). It is particularly important that mixed media should be used to support both verbal and non-verbal coding of information (**Mayer & Sims, 1994**). Thus effective web-based TBL should make good use of multimedia to enhance the pedagogic value of the content, and ILEs should encourage the use of supporting

media types.

Completely free browsing in open hypermedia systems is not usually a very efficient means of learning. The "lost in hyperspace" syndrome (**Edwards & Hardman, 1989**) has a particularly deleterious effect upon learning, principally because students are very easily diverted from their primary objectives (**McLellon, 1992**; **Dillon, 1996**; **Britt et al 1996**), resulting in what **Marchioni (1988)** called the "freedom and chaos" of hypermedia learning systems and eventually "cognitive overload". It is thus important that web-based ILEs should embody a restricted model of hypermedia that can guide students through their content and help them focus on their learning objectives. This is especially important when the students have tightly defined objectives and severe time constraints (as is often the case in undergraduate education). The interaction between student and learning environment can be considered at two levels - as "computer-user" interaction and "content-learner" interaction (**Linard, 1995**). The environment can easily have a negative impact upon student learning if it focuses on the use of the system rather than on the learning activity per se. It is important that students using the system spend their time learning rather than being trained how to use the system!

It is widely recognised by educationalists that learning usually involves several quite distinct cognitive activities. Typically these are classified as "Learning by Doing" (i.e. where students find their own way towards a goal), "Learning by Instruction" (i.e. where a teacher transmits knowledge to students) and "Learning by Exploration" (i.e. where students take the initiative to achieve a learning goal by analysing it and developing their own action plan), (**Tricot et al, 2000**). Conventional teaching almost always consists of a mixture of these three styles, and it is extremely important that ILEs make allowances for all of them. Providing that provision is made to support a choice of learning styles, then hypermedia makes a good medium for self-directed learning, because students may choose a style of interaction and engagement that suits their personal preferences (**Stanton & Barber, 1992**).

Finally, system designers should recognise that learning is ultimately a social activity, and that any attempt to isolate students (either from each other or from their teachers) is liable to damage the learning process. It is highly desirable that the functionality of the ILE should support the social dynamics of the learning community. One such model, based on a constructivist approach to learning is the Rich Environment for Active Learning (REAL) developed by Grabinger and his colleagues (**Grabinger & Dunlap, 1995**; **Grabinger et al, 1997**).

Hypermedia is extremely good at delivering both information resources and declarative exposition. However these forms of delivery constitute only one component of an effective learning process. **Laurillard (1993)** talked of a reflective dialogue between students and their teachers, where the teacher describes the central concepts and the student then responds with their understanding of said concepts. The teacher then reflects upon the student's response (and their understanding) and adapts the delivery of the concepts to suit the ongoing needs of the dialogue. Although simple in principle, this concept has yet to be fully implemented in any software. This can be seen when other TBL environments are examined, as they mostly suffer from the key pedagogic weakness of "static content", with every user receiving the same content, in the same way.

Existing Learning Environments

Much of the vision outlined above is claimed by a number of popular proprietary ILEs. However, there are a number of important problems with all of today's implementations - the most common of which is the "lock-in" issue. This is caused by the fact that most delivery frameworks depend, to a greater or lesser extent, on the

technology of the day, using proprietary interfaces and file formats for their storage and the delivery of content. This makes reusability a largely moot point. It is rare to be able to take a course, or even a subset of the information used within it, and construct a course out of it on a different system without major re-implementation. In many cases it is even difficult to reuse it for a different purpose, such as the construction of a standalone web site.

Another problem with most of these systems is their static nature, wherein no account is taken of the differing abilities and preferred learning styles of the users. This lack of personalisation can lead to students encountering inappropriate material, or appropriate material that is presented in an unsuitable manner. One major exception to this is the InterBook system developed by Brusilovsky and his colleagues (**Brusilovsky et al, 1998**). This is an experimental web-based system that uses the knowledge-based approaches of intelligent tutoring systems (**Brusilovsky et al, 1996**) to create electronic textbooks with navigation, help and guidance that adapt to suit user profiles. InterBook is capable of running adaptive courses over the WWW, however it does not support the range of pedagogic tools provided by the ILEs described above.

Another issue with most web-based ILEs is that they invariably require the use of HTML. This causes problems both to the authors, and to the reusability of content. HTML authoring tools are usually either simplistic (severely limiting the pedagogic design), or they require extensive IT/technical knowledge to use effectively (making them impractical for many non-IT-literate authors), or they produce complex HTML that is difficult to maintain or repurpose. These problems are compounded by the fact that HTML freely mixes style and content, which makes it difficult to mine, maintain or reuse this material.

The WHURLE Framework

The Components of Content - chunks, lessons & lesson plans

In WHURLE the content is discretised into atomic units called chunks. A chunk is a conceptually self-contained item that is specified in an XML file consisting of WCML (Whurle Chunk Markup Language - see Figure 1). In most cases a chunk is a small construct typically containing a single media item (e.g. a paragraph of text) or a small group of related media items (e.g. a captioned image). However, the size is not relevant per se - what is important is that it is conceptually self contained. There are cases where a single chunk may be quite large (e.g. certain legal or historical documents). An example of a simple chunk is given in Figures 1 and 2.

```xml
<?xml version="1.0" encoding="utf-8"?>
<chunk type="pic" name="genes2p3"
        title="Gene Flow" learning="ip"
        bandwidth="high" browsercap="high">

<versionlist>
        <version author="0024" date="2001-07-24"/>
</versionlist>

<media type="gif" align="center" border="0" src="cg523.gif" />

<text>
<h2>Gene Flow</h2>
<p>It has been shown that testing gene flow in an <tiny type="click"
name="arte">artificial environment</tiny> creates totally different situations
to that found in the wild.</p>
<p>The <term type="species">"tigon"</term> is the result of an artificial
cross between a tiger and a lion.  No such crosses occur in the wild, and the
hybrid is usually sterile.</p>
</text>

<tiny-text name="arte">Such as a laboratory or zoo.</tiny-text>

<keywords>
  <keyword>Gene flow</keyword>
</keywords>

</chunk>
```

Figure 1. An example of WCML. This defines a simple chunk from a biology lesson entitled "What is a Species?". This contains a single image, some text, and a tiny popup.
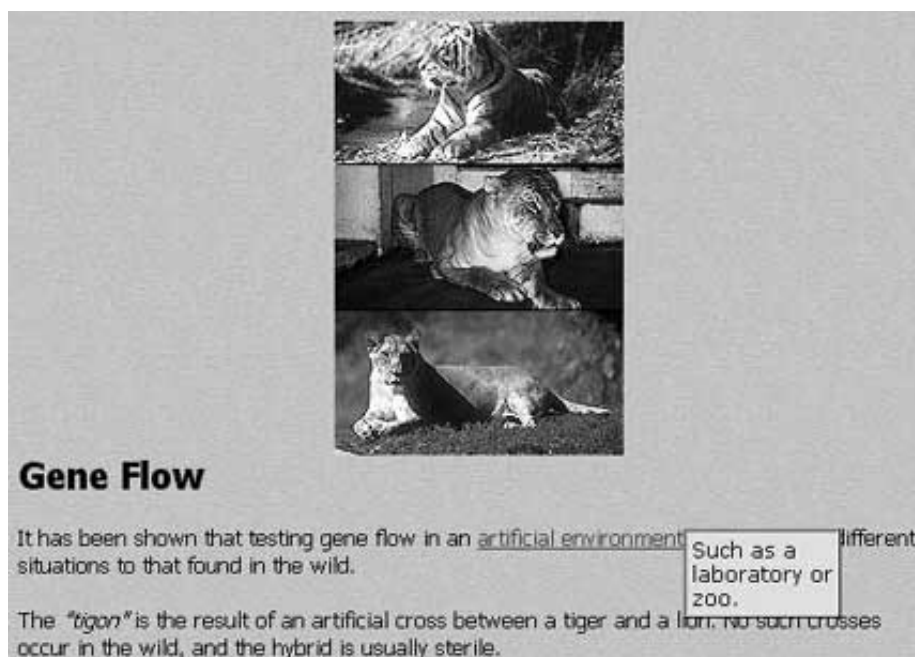

Figure 2. A simple chunk, rendered by WHURLE, that contains a single image, text and a tiny popup - the WCML that defines this chunk is shown in Figure 1.

Chunk files typically contain various metadata (in the attributes of the <chunk> element, and in the <versionlist> element), text (which is marked up using a subset of XHTML, contained in the <text> element) media (e.g. images - contained in the <media> element). MathML and SVG may be also be included in chunks - they are passed through to the browser where they must be rendered by means of native support (such as with Amaya) or by means of browser plugins. There are also tiny popups - these are small pieces of information that conceptually form a part of the chunk, but for both pedagogic and interface reasons (i.e. to avoid screen clutter) are removed from the main display. Like the text of chunks these consist of an XHTML subset, and this is contained in the <tiny-text> element. Tiny popups are linked to specific points within the text using the <tiny> element, and they are rendered as rollover popups (see Figure 2). Chunks also contain keywords, to provide a simple conceptual classification, although in the future it is intended to use a more complex ontological/semantic framework, such as RDF.

Chunks are either written by subject specialists or converted from legacy data. Authoring tools are currently being developed to facilitate chunk creation, and there are already conversion tools to create chunks from material originally created for the TLTP Biodiversity Consortium [5] . All of the chunks in any one instance of WHURLE (i.e. a single installation on a server) are collectively referred to as the melange.

The storage of content in the form of chunks is a mechanistic component of WHURLE - it is entirely transparent to the learners. What an end-user will see is a lesson, which is an apparent docuverse created by the WHURLE system. This contains the contents of any number of chunks, together with navigational links and an overlaid environment that is generated by the system. The lesson is defined by another XML file that is called a Lesson Plan, which consists of WLPML (WHURLE Lesson Plan Markup Language). A simple extract from an example lesson plan is given in Figure 3.

```
<level name="basic" title="Basic Concepts">
 <page><include xinclude:parse="xml" xinclude:href="biodvp1.xml"/></page>

 <level name="bioconcept" title="Biological Concepts">
    <page><include xinclude:parse="xml" xinclude:href="bioconp1.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="bioconp2.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="bioconp3.xml"/></page>
 </level>

 <level name="ecoconcept" title="Ecological Concepts">
    <page><include xinclude:parse="xml" xinclude:href="ecoconp1.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="ecoconp2.xml"/></page>
 </level>

 <level name="evoconcept" title="Evolutionary Concepts">
    <page><include xinclude:parse="xml" xinclude:href="evoconp1.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="evoconp2.xml"/></page>
 </level>

 <level name="genconcept" title="Genetic Concepts">
    <page><include xinclude:parse="xml" xinclude:href="genes2p1.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="genes2p2.xml"/></page>
    <page><include xinclude:parse="xml" xinclude:href="genes2p3.xml"/></page>

    <level name="breeding" title=" Captive Breeding">
        <page><include xinclude:parse="xml" xinclude:href="breedp1.xml"/></page>
        <page><include xinclude:parse="xml" xinclude:href="breedp2.xml"/></page>
    </level>
 </level>
</level>
```

Figure 3. An example of WLPML. This is an extract from a simple lesson plan, showing the hierarchy of levels and pages - the chunk "genes2p3.xml" is shown in Figures 1 & 2.

The lesson plan contains a hypermedia pathway through the melange that is created by teachers using WHURLE (although default lesson plans are provided with a melange distribution). In its simplest conceptual form a lesson plan consists of a hierarchy of levels- each containing one or more pages. Pages consist of chunks transcluded by means of Xinclude [6] . The processing of Xinclude is orthogonal to both parsing and validation, which means that chunks are retrieved as required, rather than during the parse phase. Thus there is a relatively modest processing overhead at parse time, and the server load is spread evenly during use.

When the lesson is adaptive, then the lesson plan is likely to be very much more complex than the example shown above, because the inclusion of chunk, page and even entire lesson levels is conditional - determined by dependencies in the lesson plan, as discussed below.

The Virtual Document

The virtual document is the visualisation of the content provided by WHURLE to the learners. This appears to be an interlinked web document, together with navigation and an environment that contains learning tools. This is actually an illusion created by the system to provide the end-user with a view over a part of the melange, and

the means to navigate through it. The virtual document consists of a rendering of the chunks that are defined in the lesson plan, together with navigational links automatically created from the structure of the lesson plan, and authored links defined in the linkbase. This is overlaid with the standard WHURLE environment - the cosmetic components of which are specified by a skin (an XML configuration file that specifies the interface graphics and screen layout options). An example of an end-users view of a virtual document is provided in Figure 4.
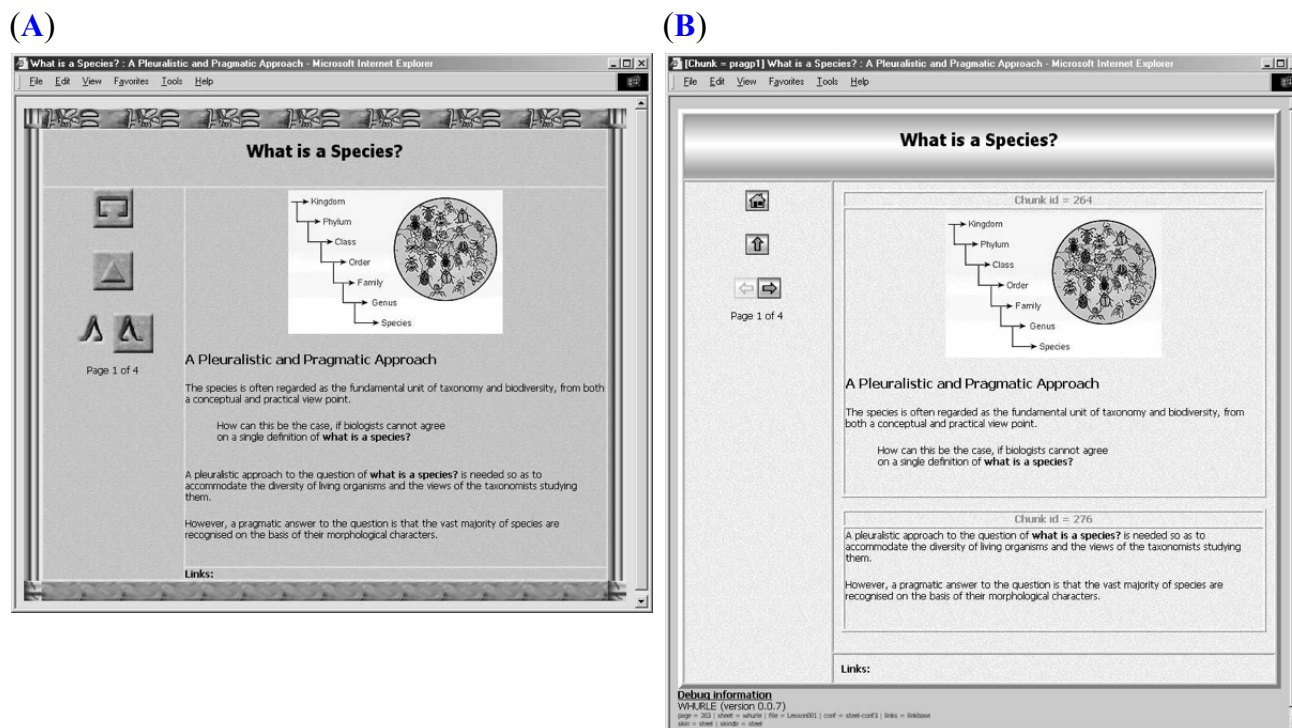
(A) (B)

Figure 4. A virtual document consisting of two chunks, rendered using two different skins (A and B). The version B is displayed with WHURLE running in a "debug mode", and thus the chunk boundaries are displayed explicitly.

The WHURLE Implementation

Most of the functionality of WHURLE is contained within a single XSLT stylesheet that is processed on the server. It is envisaged that WHURLE will always be a server based system delivering HTML (or possibly in the future XHTML) dynamically generated by the XSLT. There are a number of reasons for this design - the most pragmatic is that it is technically feasible today, using widely available open-source software, delivering to clients that can be running any browser on any platform. One major application for which we intend to use WHURLE is the delivery of distance learning material. Under such circumstances, the more proscriptive that one is about client software, the more difficult (and costly) support becomes. Also, there is already starting to be a proliferation of "thin clients", ranging from palmtop computers to browsers integrated into digital televisions. Under such circumstances, it makes sense for the XML processing to be server-side, and to deliver to a "lowest common denominator" display format (i.e. HTML). The software engine used to provide this functionality is the Cocoon XML Publishing Framework [7] , which provides processors for the XSLT and Xinclude, together with an SQL processor. The MySQL [8] database is used to store user profiles. The lesson plan, configuration file and navigational information is specified as request parameters of the URI.

Autonavigation

In the lesson plan a teacher constructs a hierarchy of pages (each consisting of one or more chunks) that can then be navigated by learners, to allow them to approach the topic in either a breadth or depth-first manner. In order to both release the teacher from the difficult challenge of constructing navigational cues throughout their material, and to allow the reuse of chunks in a position-independent manner, WHURLE uses an autonavigation system. This uses XSLT manipulations to generate the structural links that comprise the navigational components of the virtual document.

XSLT possesses a semantically rich syntax for expressing the relationship of one node to another, both in terms of ancestry and sibling (end-to-end and side-to-side) relationships. Utilizing this power we can examine the context of the current page within the lesson plan and provide cues for further movement as directed by the lesson plan. Using the standard tree-based view of hierarchically structured information we can imagine a page to be at an arbitrary position within the tree. Illustrated below (Figure 5), we see the light chunk, at level one, surrounded by others, having an ancestor (the root node), siblings and children.
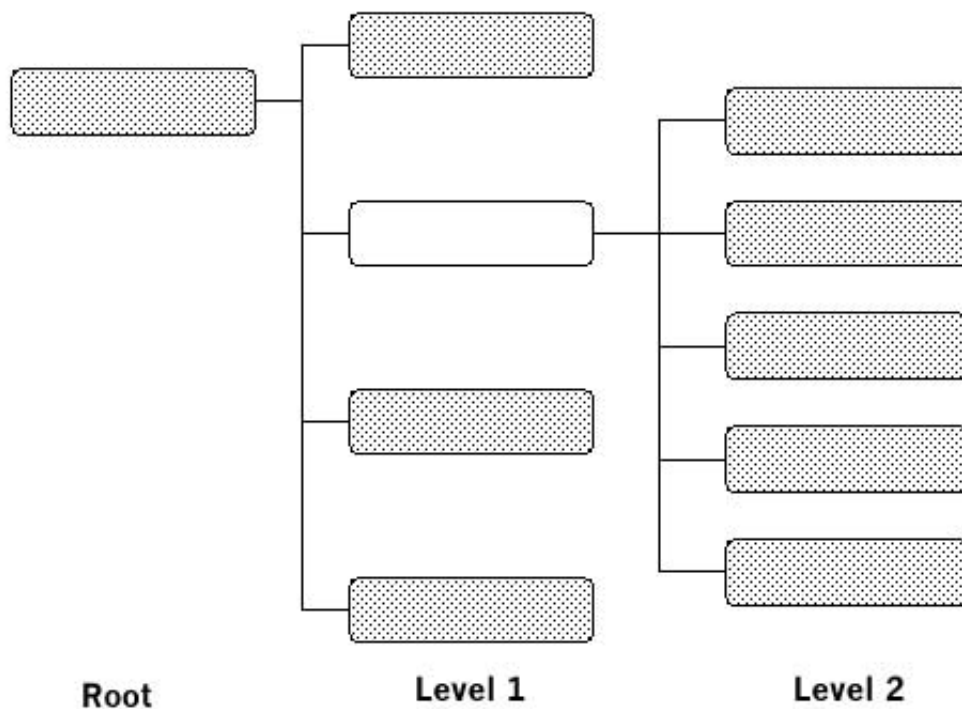

Root **Level 1** **Level 2**

Figure 5. An example of hierarchical node structure.

Using XSLT it is possible to generate a set of pointers to the current node's relations, as follows:

- **The previous nodes, back to the root, can be generated using node::ancestor operations.**
- **The descendants of a particular node can be found using descendant::NAME operations.**
- **The siblings can be found using node::siblings operations.**
- **The children can be found using child::NAME operations.**

The navigation system is based on the generation of a static, reproducible unique identifying string for the current position in the node tree - in this case an integer, n, where the node is the nth node in the overall hierarchy. This number therefore represents our current context within the node tree for XPath evaluation. Simplistically, the autonavigation then implements the following rules:

- **If the page has following siblings that are pages, we need to provide a "next page" link.**

- **If the page has previous siblings that are pages, we need to provide a "previous page" link**
- **If the page has an immediate sibling that is a level, that level and any subsequent level until the next page sibling are subtopics of the page**
- **If we are not in the first tier of <level> then we will need to be able to go up, back to the page we came from, which is not home**

The script that accomplishes this is quite complex, and so is not given here - it is, however, available on the project web site [1] .

Linking

WHURLE implements a robust system of node to node linking at the chunk level. Links are defined in a linkbase (an XML file that may be created by a teacher and specified in the lesson plan, or it may be created by the student and is a part of their user profile). In linkbase files the links are represented by the <link> element and the nodes (i.e. the link ends) are represented by the <node> element. A simple example of this is shown in Figures 6 & 7.

Links may be of three types: single, which are one to one links; hubs, which are one to many links; or plural, which are many to many links. The nodes (i.e. the ends of the links) may be of two types: internal, within the melange; or http, which are links to external web sites. The nodes of hubs have a "multi" attribute which may be set to "h" for the hub itself (i.e. the "one" end of a one to many link) or "a" for the authorities (i.e. the multiple targets of a one to many link).

```
<?xml version='1.0' encoding="utf-8"?>
<links>

<link type="single">
  <node type="internal">conser3</node>
  <node type="internal">charismatic</node>
</link>

<link type="hub">
  <node type="internal" multi="h">conser2</node>
  <node type="internal" multi="a">iucn</node>
  <node type="http" multi="a" caption="Conservation International">
        www.conservation.org
  </node>
  <node type="http" multi="a" caption="World Conservation Monitoring Centre">
        www.wcmc.org.uk
  </node>
  <node type="http" multi="a" caption="Marine Conservation Biology Institute">
        www.mcbi.org
  </node>
</link>

<link type="plural">
  <node type="internal">evoconcept2</node>
  <node type="internal">sympatry</node>
  <node type="internal">allopatry</node>
</link>
```

Figure 6. A small linkbase that defines three links. The first (the "single") is a one to one link between two internal nodes, and this is shown rendered in Figure 7A & B. The second (the "hub") is a one to many link, between internal and external nodes, and this is shown rendered in Figure 7C. The third (the "plural") is a many to many link between three internal nodes).

The links are incorporated into the WHURLE node tree at parse time using Xinclude, and the appropriate links are rendered on each page. All links are two way, and internal links do not break (i.e. if a linked chunk is removed then links to it are not rendered, although if it is reinstated then its links will still be in place.

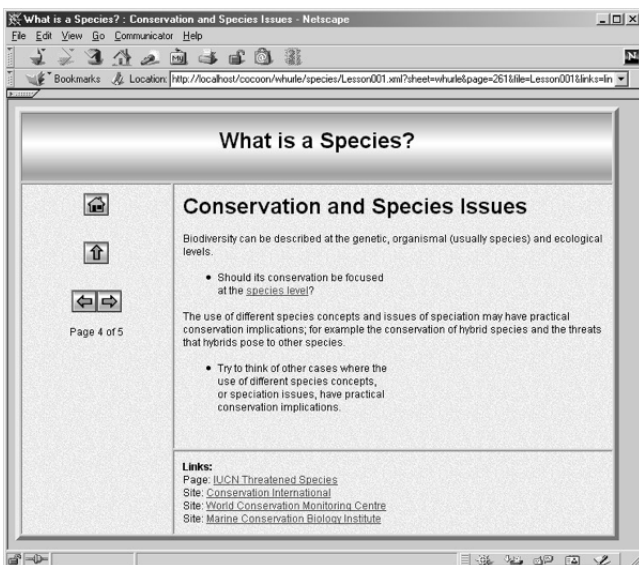(**A**)   (**B**)
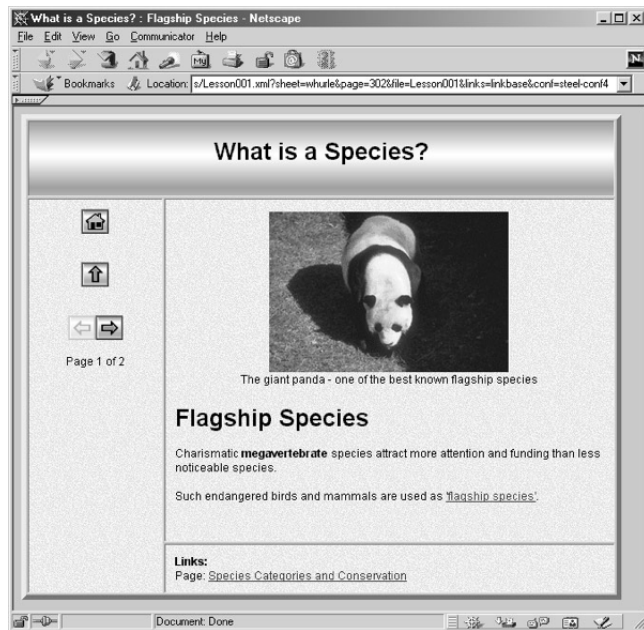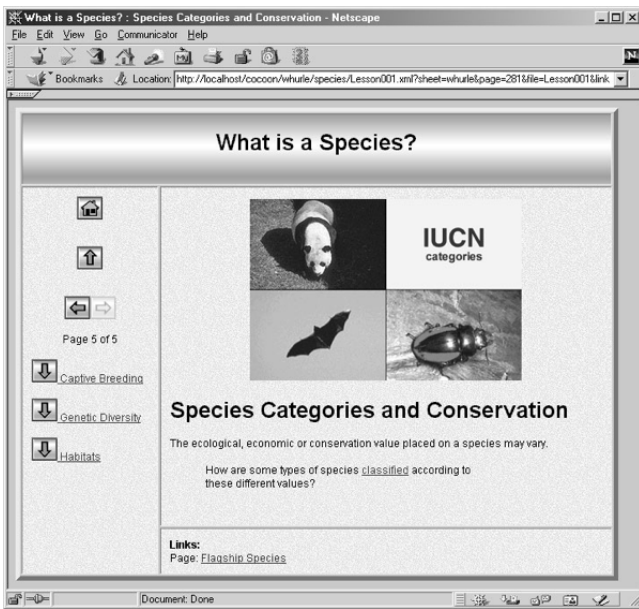
(**C**)

Figure 7. Links rendered by WHURLE - defined in the linkbase shown in Figure 6. A) is the "single" type link. B) shows the reciprocal node of this link. C) shows a "hub" linking to several internal (page) and external (site) nodes. The autonavigation system is also demonstrated here - it is manifest by the buttons to the left of the screen. A) - is the last page of a five page sequence, this page contains three child levels "Captive Breeding", "Genetic Diversity" and "Habitats". B) and C) do not have any children.

Designing for Adaptation

Lesson Plan Dependencies

A key feature of the WHURLE model, that is important for the adaptation, is that chunks, pages and levels may be included conditionally - this being determined by dependencies in the lesson plan. Dependencies are defined within a <dependancies> element which is a part of the metainformation that can be associated with lessons, levels, pages or chunks. Dependencies may involve such aspects as the learners prior experience (e.g. some lessons, levels or pages may be specified as mandatory), time (e.g. the material might only be available for a limited time period) or the ability of the learner within specified domains. The difficulty of content is specified

by an "irk" rating (i.e. the irksomeness of the material). This is then matched to personal information about the learner's abilities that is stored in the user profile. An example of dependencies is shown in Figure 8.

```
<level name="evoconcept" title="Evolutionary Concepts">
<dependancies>
    <previous_mandatory>evolution101</previous_mandatory>
    <date_available_from>2002-05-01</date_available_from>
    <date_available_until>2002-05-31</date_available_until>
</dependancies>

    <page>
     <dependancies domain="genetics" stereotype="beginner">
       <include xinclude:parse="xml" xinclude:href="evoconceptp1.xml"/>
     </dependancies>
     <include xinclude:parse="xml" xinclude:href="evoconceptp3.xml"/>
     <dependancies domain="genetics" stereotype="advanced">
       <include xinclude:parse="xml" xinclude:href="evoconceptp2.xml"/>
      </dependancies>
    </page>
</level>
```

Figure 8. An extract from an adaptive lesson plan illustrating the use of dependancies. The "evoconcept" level (entitled "Evolutionary Concepts") will be available to learners who have previously completed "evolution101", and it will be available during the month of May, 2002. Within this level the "evoconcept1" chunk will only be included if the learner's stereotype is "beginner" in the "genetics" domain (i.e. it contains introductory material). The "evoconcept2" chunk will only be included if the "genetics" stereotype of the learner is "advanced" (i.e. it contains advanced material). The "evoconcept3" chunk is always available.

Narrative in Adaptive Hypertext

The very nature of adaptive hypertext means that different learners will get different experiences from the system, and indeed the same learner might get different experiences from the system at different times. Therefore the concept of narrative needs careful consideration within this, or indeed any, adaptive environment.

The diagram shown in Figure 9 illustrates the conceptual architecture of the WHURLE system. The default narrative is determined by the combination of the lesson plan (which contains the rules for adaptation, together with a default hierarchical pathway) and the linkbase (which contains the conceptual associations of hyperlinking). The interactions of the learner with the default narrative (as visualised by the virtual document) inform the learner's user profile, which is being continually updated.
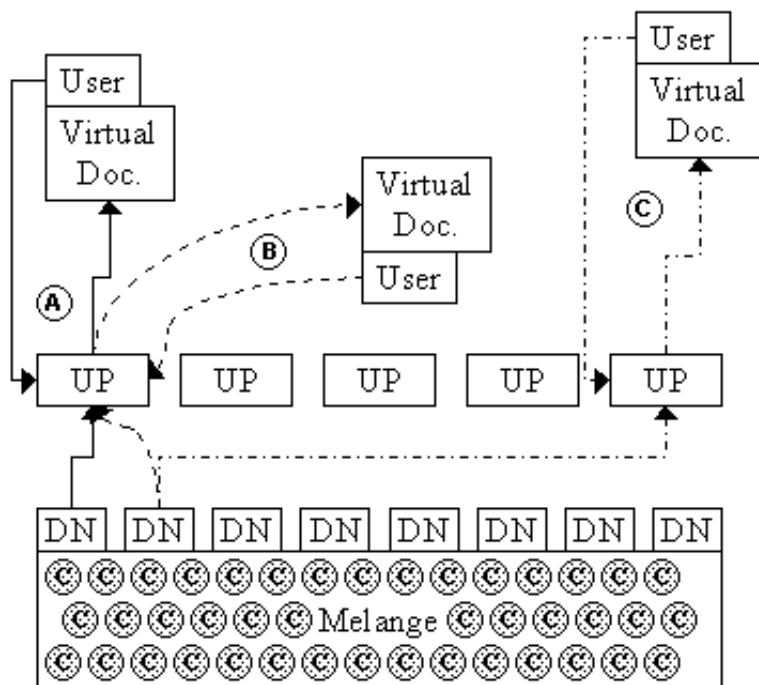
Figure 9. Conceptual model of the WHURLE system. The Melange consists of chunks (C), containing the raw content. The initial user experience is determined by the default narrative (DN) - which is itself determined by the lesson plan in conjunction with the linkbase. Adaptation of the default narrative is determined by the user profile (UP) of an individual learner, and results in the dynamic creation of a virtual document. The learner's behaviour then informs and modifies the user profile, resulting in further adaptation. The first two user experiences (A) and (B) in the above diagram represent different default narratives being used by the same learner (therefore the same user profile is updated), resulting in the creation of different virtual documents. The third user experience (C) involves one of these same default narratives but will result in the creation of a third, new, virtual document because the second user has a different and unique profile.

User Modelling in WHURLE

The user model that has been adapted for WHURLE consists of elements of two techniques that are widely used in adaptive hypermedia systems: the overlay model and the stereotype model. The overlay model measures the learner's knowledge within a given domain (**Carr & Goldstein, 1977**; **Eklund et al, 1997**), whereas the stereotype model classifies individuals according to their background and abilities (**Eklund et al, 1997**).

In the current implementation users are classified into three categories (i.e. novice, intermediate or advanced) for their knowledge level in any domain that is relevant for any given lesson plan. A lesson plan may contain any number of knowledge domains, as is appropriate to its content. Pages within an adaptive lesson may then have dependencies to target their content at learner's of certain abilities in certain domains, depending upon the scores stored in the user profile. These domain scores are independent - for example, a learner may be a novice in one domain, but advanced in another.

In the future, the plan is to extend the adaptive capability of WHURLE substantially. In addition to providing adaptation to learner abilities it is intended to also provide adaptation to their preferred learning styles. This is a non-trivial issue, because the learning styles of individuals can differ not only between knowledge domains but

also across time (**Valley, 1997**; **Groat & Musson, 1995**). In addition to the implicit adaptation of the system, users will be able to explicitly specify their own personal requirements in many areas. For example, as XML uses Unicode as its base encoding, there is not the same barrier to internationalisation and localisation as with systems using other encoding systems (e.g. US-ASCII, ISO-Latin-1). With regards to the interface, the learner will be able to change interface components (font, colours, etc.), where appropriate. All of the changes will be stored in the user profile. The infrastructure can also accommodate users with a variety of special needs. For example, output can be generated suitable for a Braille reader, voice synthesiser, etc.

Current Status and Future Directions

At the time of writing, WHURLE is in the final stages of preparation for use in teaching during the 2001-2002 academic session in multiple institutions. As such, the associated markup and infrastructures are approaching a first phase level of stability. Adaptation is working, but rudimentary. Core development continues on tuning the performance of the infrastrucure and providing the tools necessary to allow students to interact within the environment in a constructive manner.

For the future, many directions are possible. In the short term, evaluation of the experience with the first tranche of users is key to informing the future directions of the project. Key interests in the medium term include using the WHURLE infrastructure to investigate the behaviour of learners in an adaptive hypermedia environment, providing adaptation to a user's learning style and investigating the use of WHURLE as a knowledge-based information repository, tracking inquirer-expert dialogues. Adaptation should be transparent from the learner's point of view, but not from the author's. A major strand of our future research will be developing tools that allow the intuitive visualisation and authoring of adaptive content for the WHURLE environment. In the longer term, we hope to address such important issues as the automatic generation of melanges from a variety of sources, the tracking of user trust and peer review networks. We will examine the technological, social and legal ramifications of sharing and distributing melanges over pervasive and peered networks.

Acknowledgments

Endnotes

1. **WHURLE (Web-based Hierarchical Reactive Learning Environment) is an open source project (released under GPL). Further information may be obtained from the project web site at http://whurle.sourceforge.net/**
2. **WebCT (Web Course Tool) is an online course management system originally developed by the University of British Columbia, but now marketed commercially (http://www.webct.com).**
3. **TopClass is a commercial course management system targeted primarily at the industrial training market (http://www.wbtsystems.com/).**
4. **Lotus Learning Space is an educational member of the "Notes" family of products from Lotus/IBM (http://www.lotus.com/home.nsf/welcome/learnspace).**

5. **TLTP Biodiversity Consortium is a consortium funded by the HEFCE Teaching and Learning Technology Programme to develop courseware for teaching biodiversity at undergraduate level. (http://www.ibis.nott.ac.uk/biodiv/).**
6. **W3C Working Draft (2000). XML Inclusions (XInclude) Version 1.0. Available from: http://www.w3.org/TR/xinclude/**
7. **Cocoon XML publishing framework - available from: http://xml.apache.org/cocoon/**
8. **MySQL relational database - available from: http://www.mysql.com/**

References

Anderson, M. & Jackson, D. (2000). Computer systems for distributed and distance learning. Journal of Computer Assisted Learning, 16 pp 213-238.

Brailsford, T.J.; Moore, A.; Stewart, C.D.; Zakaria, M.R.; Choo, B.S.; & Davies, P.M.C. (2001). Towards a framework for effective web-based distributed learning. Tenth International World Wide Web Conference, May 1-5, 2001, Hong Kong. pp 120-121 Available via: **http://www10.org/cdrom/posters/1124.pdf**

Britt, M.A.; Rouet, J.F. & Percetti, C.A. (1996). Using hypertext to study and reason about historical evidence. IN: Hypertext and Cognition (eds. J.F. Rouet, J.J. Levonen, A.P. Dillon & R.J. Spiro) pp 43-72. Lawrence Erlbaum Associates, Mahwah, NJ.

Brusilovsky, P. (1998). Adaptive Educational Systems on the World Wide Web: A review of available technologies. Proceedings of the Workshop "WWW-Based Tutoring" at the 4th International Conference on Intelligent Tutoring Systems (ITS'98). Available via: **http://manic.cs.umass.edu/~stern/webits/itsworkshop/brusilovsky.html**.

Brusilovsky, P.; Eklund, J.; & Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference) 30 (1-7), pp 291-300. Available via: **http://www7.scu.edu.au/programme/fullpapers/1893/com1893.htm**

Brusilovsky, P.; Schwarz, E.; & Weber, G. (1996). A Tool for Developing Adaptive Electronic Textbooks on WWW. Proceedings of WebNet'96 - World Conference of the Web Society, 1996. San Francisco, CA, AACE. - pp. 64-69. Available via: **http://advlearn.lrdc.pitt.edu/its-arch/papers/brusilovsky.html**

Carr, B. & Goldstein, I. (1977). Overlays, a theory of modelling for computer aided instruction. A Technical Report. AI Memo 406, MIT, Cambridge, MA.

Davies, P.M.C. (1999). The Virtual School of Biodiversity: a model for quality assured distributed learning. IN: e-Education: Challenges and Opportunities, Proceedings of the Fifth Hong Kong Web Symposium (eds. F Castro, R Lai and Sr M Wong), Social Sciences Research Centre, The University of Hong Kong. Available via: **http://vsb.nott.ac.uk/vsb/papers/hk_conf99.html**

Dillon, A.P. (1996). Myths, misconceptions, and an alternative perspective on information usage and the electronic medium. IN: Hypertext and Cognition (eds. J.F. Rouet, J.J. Levonen, A.P. Dillon & R.J. Spiro) pp 25-42. Lawrence Erlbaum Associates, Mahwah, NJ.

Dimitriyannis, D.A. (1994). Virtual Classroom: A Case Study. Proceedings of the First World-Wide-Web conference, Geneva, Switzerland, May 25-27, 1994, pp 325-329. Available via:
http://www1.cern.ch/PapersWWW94/ddimitri.ps

Edwards, D.M., & Hardman, L. (1989). "Lost in hyperspace": Cognitive mapping and navigation in a hypertext environment. IN Hypertext: Theory into practice. (R. McAleese, Ed.), Intellect Limited, Oxford.

Eklund, J.; Brusilovsky,P.; Schwarz,E.(1997). Adaptive Textbooks on the World Wide Web. Third Australian WWW conference, Lismore NSW. pp 5-9 July 1997. Available via:
http://ausweb.scu.edu.au/proceedings/eklund/paper.html

Goldberg, M.W.; Salari, S. & Swoboda, P. (1996). World Wide Web Course Tool: An Environment for Building WWW-Based Courses. Computer Networks and ISDN Systems, 28 pp 1219-1231.

Grabinger, R.S. & Dunlap, J.C. (1995). Rich environments for active learning: a definition. . Association for Learning Technology Journal, 3 pp 5-34.

Grabinger, R.S.; Dunlap, J.C. & Duffield, J.A. (1997). Rich environments for active learning in action: problem-based learning. Association for Learning Technology Journal, 5 pp 3-17.

Groat, A & Musson, T. (1995). Learning Styles: individualizing computer based learning environments. Association of Learning Technology Journal, 3 pp 53-62.

Hutchings, G.; Hall, W.; Briggs, J.; Hammond, N.; Kibby, M.R.; McKnight, C. & Riley, D. (1992). Authoring and Evaluation of Hypermedia for Education. Computers and Education, 18 pp 171-177.

Ibrahim, B. (1994). World-Wide Algorithm Animation. Proceedings of the First World-Wide-Web conference, Geneva, Switzerland, May 25-27, 1994, pp 305-316. Available via:
http://cui_www.unige.ch/eao/www/WWW94/paper.html

Laurillard, D. (1993). Rethinking University Teaching: A framework for the effective use of educational technology. Routledge.

Linard, M. (1995). New debates on learning support. Journal of Computer Assisted Learning, 11 pp 239-253.

Mayer, R.E. & Sims, V.K. (1994). From whom is a picture worth a thousand words? Extensions of a dual coding theory of multimedia learning. Journal of Educational Psychology, 86 pp 389-401.

Marchionini, G. (1988). Hypermedia and learning: Freedom and chaos. Educational Technology, 28 pp 8-12.

Moore, A., Brailsford, T.J. & Stewart, C.D. (2001). Personally tailored teaching in WHURLE using conditional transclusion. The Twelfth ACM Conference on Hypertext and Hypermedia, August 14-18, 2001, Denmark. pp 163-164

McLellan, H. (1992). Hyper stories: Some guidelines for instructional designers. Journal of Research on Computing in Education, 25 pp 28-48

Najjar, L. J. (1996). Multimedia information and learning. Journal of Educational Multimedia and Hypermedia, 5 129-150. Available via:

http://mime1.marc.gatech.edu/MiME/papers/multimedia_and_learning.html

Quentin-Baxter, M. (1999). Quantitative evidence for differences between learners making use of passive hypermedia learning environments. ACM Computing Surveys, 31 (4). Available via http://www.cs.brown.edu/memex/ACM_HypertextTestbed/papers/52.html

Speh, M.; Butts, C.; Reilly, C. & Wang, J.C. (1994). WWW for the Globewide Network Academy. Proceedings of the First World-Wide-Web conference, Geneva, Switzerland, May 25-27, 1994, pp 335-346. Available via: http://www1.cern.ch/PapersWWW94/speh.ps

Stanton N.A. & Barber, C. (1992). An investigation of styles and strategies in self-directed learning. Journal of Educational Multimedia and Hypermedia, 1 pp 147-167.

Tricot, A. Pierre-Demarcy, C. & El Boussarghini, R. (2000). Specific help devices for educational hypermedia. Journal of Computer Assisted Learning, 16 pp 102-113.

Valley, K. (1997). Learning styles and courseware design. Association for Learning Technology Journal, 5 (2), pp 42-51.