

LINKS FOR LEARNING: LINKING FOR AN ADAPTIVE LEARNING ENVIRONMENT.

A. Moore, C. Stewart, D. Martin, T. Brailsford, H. Ashman

School of Computer Science and Information Technology, University of Nottingham,

Nottingham, U.K. +44 115 951 4231

{axm|cds|djm|tjb|hla}@cs.nott.ac.uk

ABSTRACT

The automated generation of links within a body of educational material is critical to the cost-effective implementation of Technology Based Learning. This paper describes the linking system used by WHURLE - an adaptive web-based integrated learning environment. Structural links are dynamically generated from lessons using an autonavigation system. Authored links are created by teachers or students, and consist of bidirectional links between different points in the content or between WHURLE and elsewhere on the Web. Authored links are defined in a linkbase that is either attached to a lesson (i.e. teacher originated) or to a user profile (i.e. student originated). These links may be single (i.e. one to one), hubs (one to many) or plural (many to many), and are implemented by Goate, a content modifying proxy system.

Keywords

Adaptive Hypertext, Proxy based Linking, Learning Environment, Education

1. INTRODUCTION

The web-based delivery of learning materials has become an important component of many types of education at almost all levels. While this is often driven by growing class sizes and declining resources, if technical implementations are driven by sound pedagogy, hypertext can enrich the learning experience.

Adaptive hypertext can deliver content that is personalised to the requirements of individual students (e.g. [9], [14]). Students are diverse in terms of their experience, expectations, abilities and interests, and they vary in the learning styles that are most efficacious for them. The static links widespread in Technology Based Learning (TBL) software provide the same experience for every user, regardless of abilities or interests. Adaptive hypertext attempts to overcome this limitation by selecting content to personalise the experience. However, experimental work [22] shows that hypertextual learning materials can polarise students achievements – some perform better, but others are disadvantaged. In the context of this work we have developed an experimental Integrated Learning Environment called WHURLE (Web-

based Hierarchical Universal Reactive Learning Environment) [20] [7], an XML-based adaptive hypertext system. This paper discusses issues and solutions to linking in educational hypertext.

1.1 Linking for Educational Content

Linking is critical to the design of educational hypertext systems, as one of the most important benefits of this approach to education is its ability to guide students through a vast corpus of learning resources via pathways. There are dangers associated with this – students' study-time is invariably limited and the Web has many distractions and questionable material. A student needs practice and skill to make effective use of their time in this environment. It usually falls to teachers to provide guidance, and to authoring and delivery tools and learning environments to assist teachers with this. Linking allows teachers to provide students with a structured experience, flexible enough to cater for a variety of needs but constrained enough to fit within a curriculum. Therefore, learning environments should support many link types, with such features as bidirectionality (i.e. links that can be followed in either direction) and robustness (i.e. links that don't break).

This is an area where many learning environments fail to deliver. It is rare for links in web-based systems to be more than simple and homogenous one-way pointers to resources that may, or may not exist.

Another issue with linking in educational material is that it must be easy to create and maintain links, so that it eases teachers' workloads, rather than adds to it. The key to this is in the computation of links. It is generally accepted that automatic link generation is important for the creation and maintenance of hypertext (e.g. [1] [4] [24]), especially where the underlying content is frequently changing [3] [13]. This is clearly useful for structural links, such as those used to navigate around a well-defined body of material (i.e. autonavigation), or for links that constitute a part of the learning environment (e.g. a contextual help system or discussion forum, or a glossary of relevant terms). In many cases large numbers of links can easily be generated using a simple computation (e.g. for links from names of historical characters to a bibliography). However, it is still important for teachers to be able to insert links by hand,

either to their own material or to distributed resources. These are *authored* links, and they enable teachers to provide guidance to the use of distributed resources. In addition to authored links being created by teachers, it is also important that students have the capability of creating their own links as this allows for a constructivist educational model. This might be for study or revision, or for publishing to other students in collaborative learning, or as part of work assignments. Both of these types of links have to be hand authored in WHURLE as the web-form used for content authoring does not support their creation. Section 3.3 discusses these links further.

1.2 Open Hypertext Systems

In open hypertext systems, such as the Distributed Link Service [12], link services are provided by an engine that is separate from both the data and the software that renders that data for viewing. Hypertext functionality is made available to any application, without the need for that application to itself store the links [6]. Links (or the computation specifications for computed links) are stored in a linkbase separate to the data being linked, allowing linking of arbitrary content from arbitrary sources [25]. The principles of open hypertext are important to the design of effective adaptive learning environments, facilitating the dynamic modification of the ordering and presentation of content dependent upon user profiles. It also makes it feasible for students to have their own linkbase, which they can create and modify, providing both an ‘ownership’ of content created in this method, and a personalized resource.

2. THE WHURLE FRAMEWORK

WHURLE is an experimental adaptive learning environment [7] [27] which stores information as atomic units, called chunks. These are the smallest possible conceptually self-contained units of information that can be used by the system. They may be as small as a captioned image or a single paragraph of text, or they may be as large as entire legal or historical document. Lessons consist of a collection of chunks, together with a default pathway, or lesson plan, defined by teachers. The lesson plan is filtered by an adaptation filter that implements the user model based upon data stored in the student’s user profile.

2.1 Chunks, Lessons and the Virtual Document

All WHURLE data is stored as XML files, using purpose-designed markup languages (chunks use the WHURLE Chunk Markup Language – WCML, and lesson plans use the WHURLE Lesson Plan Markup Language – WLPML). Chunk files typically contain various metadata, text (which is marked up using a subset of XHTML) and other

media. MathML and SVG may be incorporated into chunks although they must be rendered by means of native support in the browser (e.g. with Amaya) or by means of plugins. Chunks are created either by subject specialists, or generated from legacy data using purpose-built tools.

Chunks are not something that students see directly - they see a lesson. A lesson is an apparent docuverse that contains material defined in chunks, together with navigational links and an overlaid environment – both of which are generated dynamically by the system. In their simplest form lesson plans consist of a hierarchical structure, with each level containing one or more pages. Pages contain one or more chunks, the content of which is incorporated into the XML node-tree at parse-time by means of XInclude. In adaptive lesson plans, the data is more complex because the conditional inclusion of components of the lesson is determined by dependencies. To the student, WHURLE provides an environment in which a hyperlinked document is delivered. This virtual document is an illusion - it is an end-user view provided over that part of the total content that is currently being presented, with navigational (for example, navigational linking modifications can be caused due to adaptation) and interface (for example, an interface designed for colour blind users) overlays on top.

3. Linking in WHURLE

WHURLE contains three distinct linking systems, serving distinct purposes. The first category is the *intra-chunk links* (i.e. links within individual chunks) that are created by the original chunk author, and form a part of the chunk. Secondly, *systemic links* are automatically generated by the system to provide navigational facilities and the user interface of the learning environment. Finally *authored links* are manually created, by teachers or students, and are stored in one or more linkbases, separate from the content.

3.1 Intra-chunk linking

Sometimes it is necessary to provide some form of linking within a chunk. This is when a small piece of information is conceptually a part of a chunk, but for pedagogic or user interface design reasons it needs to be removed from the main display. An example might be an answer to a question. The answer should be part of the same chunk as the question (the answer makes no sense without the question), but it may not be desirable to display the answer until after the student has had a chance to think about it. Another example might be where there is a lot of textual information in a chunk, and displaying it all at once would result in screen clutter. Intra-chunk links are currently implemented as small rollover pop-ups that contain content which – like the text of chunks – consists

of an XHTML subset. These elements are linked to specific points within the chunk (e.g. attached to a specified text range, or attached to co-ordinates of an image).

3.2 Systemic links

Systemic links are generated by the system, and create both the virtual docuverse and the learning environment (i.e. the student’s personalised view of the content). These links do not operate at the level of chunks, but provide the higher level navigational and environmental overlays. They are generated automatically at parse time – in the case of the navigational overlay this is based upon the structure of the lesson plan, but in the case of the environment it is based upon definitions in the skin (an XML file that determines the user interface). The autonavigation system depends on the hierarchy of pages within the lesson plan. The chunks themselves can be reused in a position-independent manner, and teachers do not need to insert or maintain navigational links in their material. This is implemented by XSLT manipulations of the node tree to allow the visualisation of ancestral, sibling and child relationships. The context of the current page within the lesson plan is determined, and links are then generated to enable future movement within the apparent docuverse. The skin defines most cosmetic aspects of WHURLE – such as the graphics that constitute the user interface to learning support tools and a help system.



Figure 1. Linking in WHURLE - this demonstrates intra-chunk linking (the pop-up), systemic linking (autonavigation, resources, system operations) and an authored link (the link to Apache).

3.3 Authored links

The key requirement for authored linking is that teachers or students should be able to create links to any available resources. These links may be to other parts of the WHURLE lesson or to anything on the Web. These links are defined in a linkbase, which is an XML file that is

either created by a teacher and specified in the lesson plan, or created by the student and stored as a part of their user profile.

Each link in the linkbase contains nodes (link ends). WHURLE links are single (i.e. one to one links), hubs (i.e. one to many links) or plural (i.e. many to many). Within the hub link type the nodes are specified as either the hub itself (i.e. the “one” end of a one to many link) or an authority (i.e. the multiple targets of a one to many link). Links are all two way, and are designed in such a way that they do not break - if a linked item is removed, then any links to it are not rendered, although if the item is reinstated then its links will still be in place.

Figure 2 shows a simple WHURLE linkbase, illustrating the three different link types.

```
<?xml version='1.0' encoding='utf-8'?>
<links>
  <link type="single">
    <node type="internal">intro006</node>
    <node type="http" caption="Apache – an open source server">www.apache.org</node>
  </link>
  <link type="hub">
    <node type="internal" multi="h">intro004</node>
    <node type="http" multi="a" caption="World Wide Consortium">www.w3.org</node>
    <node type="http" multi="a" caption="Netscape – a browser">www.netscape.com</node>
    <node type="http" multi="a" caption="Mozilla – a better browser">www.mozilla.org</node>
  </link>
  <link type="plural">
    <node type="internal">intro002</node>
    <node type="internal">intro008</node>
    <node type="internal">intro024</node>
  </link>
</links>
```

Figure 2. A small linkbase that defines three links. The first (the “single”) is a one to one link between internal and external nodes (see Figure 1). The second (the “hub”) is a one to many link. When the hub page is displayed, the links to all the authority pages are displayed, if an authority page is displayed, only one link, back to the hub, is displayed. The third (the “plural”) is a many to many link between three internal nodes. Any page where any of the chunks in the plural is shown contains links to all other pages where the other chunks are found.

3.4 Proxy based linking using Goate

Links can either be hand-made, or they can be generated by a link computation agent, and then stored in the linkbase. WHURLE has no inbuilt link computation

capacity but instead farms out this requirement to a content modifying proxy, Goate [19]. WHURLE essentially acts as a content management system - link specifications are generated by WHURLE but interpreted by Goate, which provides the emulation of the advanced WHURLE linking specification in the restricted internet (HTML) linking environment. Goate operates as a HTTP proxy. As a proxy, Goate, intercepts both requests for pages and responses given by servers. Whilst normally a proxy passes back content unaltered, Goate can alter the pages before they reach the browser. A similar approach has been previously discussed and implemented with systems such as the DLS [5] [8] [11] [16] [18].

3.4.1 Language modules

Goate uses plugins to enable 3rd party linking language support. Any developer with a copy of Goate can develop support for any linking language using the Goate API. The plugins to support new linking languages are 'language modules', and are loaded at run-time and dynamically linked into the proxy.

Each language module is responsible for the declaration of relevant links. That is, given the page being viewed, the language module identifies links of its supported language that apply to this page and declare them to Goate. The links may have been obtained from the document source

(e.g. recognition and parsing of an embedded XLink) or declared as the current page matches an end-point in a link-base. The language module need not handle the synchronisation of end-points of bi-directional links, but needs only to supply a method to re-evaluate an end-point position if Goate detects the change in a page it was based on.

3.4.2 Link homogenisation

Considering a HTML document, some of the content will signify links. For example, a <a href> tag. Whilst <a> is considered a link as per the HTML specification it is possible to consider any tag, or section of text a link if the linking language calls for that. So, in some instance we might say "form a link from every piece of bold text to this page". We consider the <a> tag to have a purely linking purpose, whilst the tag does not, i.e. even if tags are not links in a document, there is still a reason for them to be present. All links for a page are stored in a common format, then purely-linking effect tags are removed to stop their linking effect clashing with the links we add (if a tag has a linking effect in the environment but not a 'pure' linking effect, it's linking effect must be neutralised in some way.). Once we have all the links in this common format it is easier to rationalise them to the environment (e.g. handling overlapping).

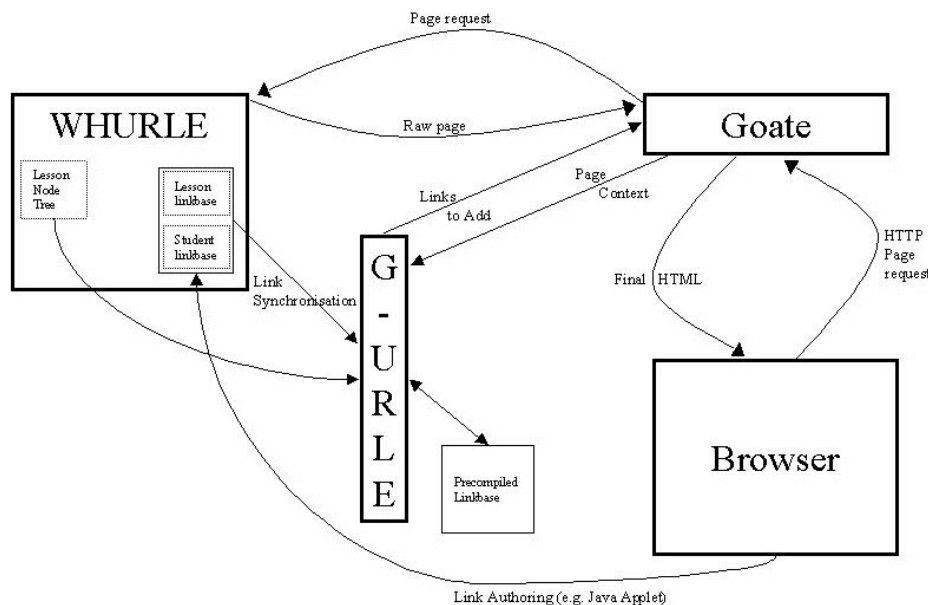


Figure 3. Ghurle System Diagram. WHURLE is accessed via the Goate proxy. The browser passes a request to the proxy, which in turn passes a request to WHURLE, which returns a 'raw' page to Goate. The Ghurle language module adds the links salient to the raw page from the precompiled linkbase, in relation to the current lesson node tree. The links are rendered using the appropriate rendering module (not shown) and then passed back to the browser. Authored links may be added to the student linkbase by, for example, a java applet tool provided by the WHURLE interface.

The rationalised links are then added back into the environment, in the case of HTML this means adding <a href> tags around sections. The presentation attributes of these links, such as colour, style, and so on is a separate issue. The presentation attributes are defined as a composite of preferences from the part of the system which recognised the links and any user preferences.

3.4.3 Rendering modules

Goate uses plugins for the rendering stage of the process, that is, for going from an HTML document and a collection of links to the modified HTML. Since different browsers have different capabilities (regarding JavaScript, CSS etc.) and those that share capabilities often have different implementation standards, a modular approach here provides abstraction and allows 3rd party support for a particular browser.

Goate loads all ‘rendering modules’ present in a system at startup. When a HTTP request is made, Goate uses the user-agent string sent by the browser to decide which rendering module to use. Each rendering module is sent the user-agent string and will return if it can: output tailored for this particular browser or a default (text-only) rendering which will probably be OK for the browser. The first offer of tailored output is accepted, if none are available the first default rendering is accepted. The results of the rendering module selection are cached, so future requests from the same browser do not go through the selection process repeatedly.

3.5 The Goate-WHURLE Interface

Utilising Goate as the linking engine for creating computed links for WHURLE involves the use of the Goate-WHURLE language module (Ghurle). This module pre-fetches and compiles the linkbases used by the current lesson, and stored in the current learner’s profile. For each page fetched from the system, the relevant links from the linkbases are inserted into the link list at the bottom of the page, compared against the current node tree of the lesson. This allows adaptive linking, where links to material not currently available to the learner are treated separately (currently simply not displayed). To do this, Ghurle declares links with the “triggers” at the bottom of the relevant page. These are declared around a span (perhaps a zero size point), and the trigger presentation style replaces the selected span with the link title.

4. Related Work

There are now many Integrated Learning Environments in common use – some, such as CT [17] and COSE [23] have been developed by the academic community, whilst others such as Blackboard¹, Virtual-U², Lotus Learning Space³ and Topclass⁴ are commercial products. CT⁵ has made the transition from academia to one of the most

widely used of these commercial systems. Although they differ in detail, these systems are all essentially educational content management systems for the web. They all provide robust systemic linking with autonavigation being overlaid on top of content - usually stored in an underlying database. They also contain a wide variety of learning tools – ranging from collaborative work and discussion areas to glossaries and annotation systems. Adaptive Integrated Learning Environments are rather less common, and have mostly been developed as research projects. Some of the better known are AHA! [15], CHEOPS [21] and Interbook [10]. These too have robust systemic linking systems, although unlike the systems mentioned earlier their content is adaptive – and so either the content or the links (or both) are modified according to the user model that they are implementing. The linking requirements of WHURLE are slightly different, owing to the atomistic nature of WHURLE data and the use of conditional transclusion as the main adaptation mechanism [20]. The highly constrained intra-chunk links of WHURLE are needed because of the self-contained nature of chunks (which is fundamental to the adaptation model of WHURLE). All of these systems provide the capability for manually inserted links – what we have called authored links. Many of these systems make some provision for the maintenance of link integrity and bidirectionality – for links within the system. However, for authored links to external sites all that is usually available are the usual one way pointers of HTML. The mechanics of the system do not lend themselves to robust bidirectional linking, but content modifying proxies, such as Goate, do provide a means of achieving this.

5. Conclusions

Currently linking within TBL software is a time-consuming process, both for authoring and maintenance. By keeping authored links separate from the main body of informational content of WHURLE, these problems are alleviated, allowing the creation and curation of both intra- and extra-system links as a separate process, and allowing learners to build their own relationships within the system. The combination of WHURLE and Goate utilises open hypertext principles, adaptive hypertext principles and computed linking, resulting in a linkbase much simpler to create and maintain, with Goate as the linking engine computing links, adding them to the existing teacher-created links stored by WHURLE.

6. Commercial ILEs

1 Blackboard - <http://www.blackboard.com/>

2 Virtual-U - <http://www.vlei.com/>

3 Lotus Learning Space - <http://www.lotus.com/home.nsf/welcome/learnspace/>

7. Acknowledgements

We should like to thank Mohamed Ramzy Zakaria and the other members of the Web Technology Group for many useful discussions, as well as their unstinting support. We are also grateful to the EU Minerva programme for funding under the Adapt project.

8. References

- [1] Agosti, M. Allan, J. (eds) (1997) Special Issue on Methods and Tools for the Automatic Construction of Hypermedia, Information Processing and Management, 33(2), Elsevier.
- [3] Ashman, H.L., (2000). Electronic Document Addressing - Dealing with Change, 32 (3) September 2000, ACM Computing Surveys.
- [4] Ashman, H., Garrido, A. Oinas-Kukkonen, H., (1997). Hand-made and Computed Links, Precomputed and Dynamic Links, Proceedings of Hypermedia - Information Retrieval - Multimedia '97 (HIM '97) conference, September 1997, pp 191-208.
- [5] Barrett, R. Magilo, P.P. (1998). Intermediaries: New places for producing and manipulating content. Seventh International World Wide Conference (7), Brisbane, Australia, 14-18th April 1998.
- [6] Brailsford, D.F. (1999). Separable hyperstructure and delayed link binding, Computing Surveys, Symposium Issue on Hypertext and Hypermedia (eds H. Ashman and R. Simpson), ACM, 31 (4es).
- [7] Brailsford, T.J.; Stewart, C.D.; Zakaria, M.R. Moore, A. (2002). Autonavagation, Links and Narrative in an Adaptive - Based Integrated Learning Environment. 11th Intl. World Wide WebConference (2002), Hawaii, May 2002.
- [8] Brooks, C.; Mazer, M.S.; Meeks, S. Miller, J. (1995). Application-specific proxy servers as HTTP stream transducers. Fourth International World Wide Conference, (4), Boston, Massachusetts, December 11-14, 1995.
- [9] Brusilovsky, P. (1998). Adaptive Educational Systems on the World Wide : A review of available technologies. Proc. Workshop "Web Based Tutoring" at the 4th International Conference on Intelligent Tutoring Systems (ITS'98).
- [10] Brusilovsky, P.; Eklund, J. Schwarz, E. (1998) based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems (Proc. 7th Intl. World Wide Web Conference, April 1998) 30 (1-7), 291-300.
- [11] Carr, L.A.; DeRoure, D.; Hall, W. Hill, G. (1995). The distributed link service: A tool for publishers, authors and readers. Fourth International World Wide Conference, Boston, Massachusetts, December 11-14, 1995.
- [12] Carr, L.A.; Hall, W. De Roure, D. (1999). The evolution of hypertext link services, Computing Surveys, Symposium Issue on Hypertext and Hypermedia (eds H. Ashman and R. Simpson), ACM, 31 (4es), 1999.
- [13] Davis, H., (1998). Referential integrity of links in open hypermedia systems. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia (Hypertext '98). pp 207-216.
- [14] De Bra, P.; Brusilovsky, P. Houben, G.J. (1999). Adaptive hypermedia: from systems to framework, Computing Surveys, Symposium Issue on Hypertext and Hypermedia (eds H. Ashman R. Simpson), ACM, 31 (4es).
- [15] De Bra, P.L. Calvi (1998). AHA! An Open Adaptive Hypermedia Architecture, The New Review of Hypermedia and Multimedia, 4, pp. 115-139.
- [16] De Roure, D.; El-Beltagy, S.; Carr, L. Hall, W. (1999). A Distributed Link Service using query routing. Eighth International World Wide Conference, (8) May 11-14, 1999, Toronto.
- [17] Goldberg, M.W. Salari, S. (1997). An Update on CT ed - Based Learning Environments. Proceedings of NAU '97 - Current Practices in -Based Course Development, June 12 - 15, 1997, Flagstaff, Arizona.
- [18] Grønback, L.S.K. Ørbæk. P. (1999). vise: browser and proxy support for open hypermedia structuring. Eighth International World Wide Conference, (8) May 11-14, 1999, Toronto.
- [19] Martin, D. Ashman, H. (2002). GOATE: XLink and Beyond. Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (Hypertext '02), pp 142-3.
- [20] Moore, A., Brailsford, T.J. Stewart, C.D. (2001). Personally tailored teaching in WHURLE using conditional transclusion. Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext '01). pp 163-164.
- [21] Negro, A.; Scarano, V. Simari, I.L. (1998). User Adaptivity on through CHEOPS. Proc. 2nd Workshop on Adaptive Hypertext and Hypermedia, held at the 9th ACM conference on Hypertext and Hypermedia (Hypertext '98), Pittsburgh PA, June 20-24, 1998.
- [22] Quentin-Baxter, M. (1999). Quantitative evidence for differences between learners making use of passive hypermedia learning environments, Computing Surveys, Symposium Issue on Hypertext and Hypermedia (eds H. Ashman R. Simpson), ACM, 31 (4es).
- [23] Roach, M.P. Stiles, M.J. (1998). COSE - A Virtual Learning Environment founded on a Holistic Pedagogic Approach. CTI: Software for Engineering Education 14, pp 5-11.
- [24] Thistlewaite, P. (1997). Automatic Construction and Management of Large Open Webs, in Agosti and Allan (eds), Sp. Issue on Methods and Tools for the Automatic Construction of Hypermedia, Inf. Proc & Mgmt, 33(2) pp 161-173.
- [25] Verbyla, J.L.M. Ashman, H.L. (1994). A User-Configurable Hypermedia-based interface via the functional model of the link. Hypermedia 6(3), pp 193- 208.
- [26] Zakaria, M.R. Brailsford, T.J. (2003) User Modelling and Adaptive Educational Hypermedia Frameworks, to appear, New Review of Hypermedia and Multimedia (Special Issue on Hypermedia and the Web), Taylor- Graham.
- [27] Zakaria, M.R., Moore, A., Stewart. C.D., Brailsford, T.J. (2003) "Pluggable" User Models for Adaptive Hypermedia in Education. Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia (Hypertext '03). pp 170-171.