Giving learners a real sense of control over adaptivity, even if they are not quite ready for it yet.

Marek Czarkowski and Judy Kay
School of Information Technologies
University of Sydney
Australia, 2006
{marek,judy}@it.usyd.edu.au

Abstract

This chapter describes Tutor3, the latest in a sequence of systems that we have created to provide adaptation of hypertext where the user can maintain a real sense of control over the adaptivity. In Tutor3, the user always has access to precise details of *what* has been adapted to them, *how* this adaptation is *controlled* and they can *alter* it. We describe both the user's and the hypertext author's view of the system. We then report a qualitative evaluation of the system in terms of the ways that users were able to understand both the adaptation and their power to control it. We conclude that while users do expect adaptivity, they do not expect to be able to control it. We discuss the challenges this creates for building adaptive systems that users can control effectively.

Introduction

Adaptive hypertext, at its best, offers the promise of a personalised document and interaction that meets the individual's particular preferences, knowledge and goals. There are many situations where this could be of immense value. For example, consider the case of hypertext learning environments. These offer potential improvements in learning outcomes if they deliver some of the benefits that appear to be achievable in one to one tutoring (Bloom, 1984). Equally importantly, users who have sensitive information needs may appreciate it if this is personalised. For example, in an evaluation of adaptive presentation of information for patients with cancer (Cawsey, Jones and Pearson, 2000), there was a strong preference for the adapted version of the information.

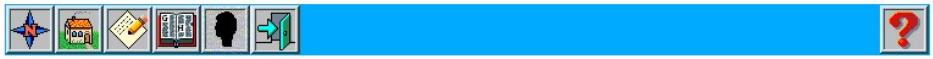
While personalisation has the potential to offer considerable benefits, it also has some serious problems. In this chapter, we are particular concerned with one class of these. They are associated with the potential for adaptive systems to be unpredictable and irritating because the user is unable to determine what is adapted, how that adaptation is controlled and how they can manage the personalisation processes. Users may be surprised or irritated by systems that are 'too smart'. Users may be subject to the hunting problem, where the system and the user simultaneously attempt to adapt to each other (Browne, Totterdell, Norman, 1990). It is also quite possible that the author of the adaptive hypertext system has made a mistake, such as providing copious detail when the user has asked for minimal information. This could be due to a simple coding error where a single 'not' was omitted, or incorrectly included.

To delve into this issue, we first need to identify the core elements of an adaptive hypertext. While these will vary across systems, they would generally include the following four elements.

- A user model is an essential part of an adaptive hypertext system since it is the system's knowledge of the user and is the driving force determining exactly what is adapted and how. The user model may be very simple, perhaps a set of Boolean flags or an arbitrarily complex representation.
- The adaptable content. This might be as simple as text snippets, each of which is either selected or not for a particular user. At the other extreme, it may be a complex knowledge representation.
- The adaptation process which combines the above two elements to produce the adaptive presentation.
- The user modelling process which determines the user model and its evolution over time. Like all the other elements, this can range from a very simple form, such as the user always setting the values of some flags in what is often called customisation. At the more complex end of the spectrum, it may involve machine learning, based on information collected about the user or it may involve information and knowledge about other users such as stereotypic users (Rich 1979) and deep knowledge of the domain.

Each of these constitutes one part of the process of producing an adaptive hypertext. Each could be responsible for presenting a different hypertext to different users. If the user is surprised at what they see in such a hypertext environment, it might be due to an error in any of these elements. Even if there is no error, a user might wonder why the system presented the information that it did. They may also wonder what other people would see that they did not. If they watch over the shoulder of another user of the same adaptive hypertext system, they may well be surprised at differences in the way the system treated that user compared with the way it treated them. If the user model was defined some time ago, the user may have completely forgotten the details of the set up. In that case, if the user has changed over time and the system does not model that change, its adaptation could be wrong and increasingly so. If users choose to share login accounts, they may not realise the impact this has on adaptation. We have been working to explore ways to address these problems by making adaptive hypertext that is *scrutable*, by which we mean that the user can delve into each of the elements of the adaptation to see what it is doing and we would like to support the user in controlling at least some of these elements.

Consider, for example, the two pages shown in Figures 1 and 2. These are two of the possible adapted forms of a single web page in our adaptive hypertext system called Tutor. In a typical adaptive hypertext system, the user who is presented with the page shown in Figure 1 has no way of knowing that another user would see the form in Figure 2. Further, the user would not be easily able to determine why the system chose to adapt in the way that it did. A critical difference between Tutor and more conventional hypertext systems is indicated by the link at the bottom left. This *How was this page adapted to you?* link enables the user to scrutinise the processes underlying the adaptation.



topic: UNIX2 > 1.0 Introduction to UNIX > 1.3 Directories

1.3 Directories

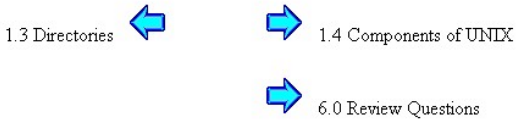
Computers can store large amounts of data, and therefore many files. In order to manage the files we store on a computer we need to be able to organise them.

UNIX, like most operating systems, uses the concept of directories to allow the user to organise files. A directory is a named collection of files. The user is able to group files in directories, and refer to these directories by name. The UNIX directory structure is heirarchical, meaning that directories can further be grouped in directories. UNIX enables the user to create, delete, move and rename directories, aswell as allowing the user to traverse the directory structure.

The top level directory in a UNIX system is called the root directory and is refered to in UNIX commands using the single '/' character.

By convention, UNIX systems usually have the following standard directories stemming from the root directory:

Directory	What it is used for
/home	Used to store the home directories of each user. Each user stores their files mainly under their own home directory, thus keeping their files seperate from other users.
/usr/bin	Stores common user programs.
/tmp	Used to store temporary files. This directory usually emptied when the system is rebooted
/dev	Contains device drivers.
/etc	Conatins files used for system adminsitration, such as the password file /etc/passwd.
/kernel	Contains programs and files which comprise the system kernel engine.



How was this page adapted to you?

Figure 1. Example of a page adapted to a student who wants to learn as much as possible.

topic: UNIX2 > 1.0 Introduction to UNIX > 1.3 Directories

1.3 Directories

Computers can store large amounts of data, and therefore many files. In order to manage the files we store on a computer we need to be able to organise them.

UNIX, like most operating systems, uses the concept of directories to allow the user to organise files. A directory is a named collection of files. The user is able to group files in directories, and refer to these directories by name. The UNIX directory structure is heirarchical, meaning that directories can further be grouped in directories. UNIX enables the user to create, delete, move and rename directories, aswell as allowing the user to traverse the directory structure.

The top level directory in a UNIX system is called the root directory and is refered to in UNIX commands using the single '/' character.

How was this page adapted to you?

Figure 2. Same page as in Figure 1 but this time adapted to a student who wants to learn just enough to pass.

The adaptation performed by Tutor can be coarse-grained or fine-grained. In the above example, a whole paragraph has been adapted. However, adaptive content can range in size, be it a whole page, paragraph, word, or even a single HTML element. This gives the author flexibility as they can write adaptive words mid-sentence rather than providing two versions of the paragraph which would otherwise be the same. Since HTML browsers support different forms of media (text, sound, image), it is possible to adaptively include or exclude these forms of media based on the user's profile.

Tutor is a generic framework for the presentation of adaptive web based course material. Tutor can host multiple courses, each with its own set of adaptive lesson pages. To define a new course, the author writes the learning material as a set of documents in the Adaptive Tutorial Mark-up Language (ATML). An ATML document is essentially an HTML 4.0 document with additional mark-up that is used to describe adaptive content. Next the ATML documents are uploaded to the web server where the Tutor web site is hosted and the course is ready to be accessed by students. All this makes Tutor quite unexceptional as an adaptive hypertext system as we use well-known techniques for adaptive content presentation and adaptive navigation. The distinctive aspect of our series of Tutor systems, and the focus of this paper, is that Tutor represents an exploration of how to build scrutability into adaptive hypertext.

The next section of this chapter provides some background to our motivations for building a scrutably adaptive hypertext and some of the work that has influenced the design of Tutor3. Next is a brief overview of some of the user's view of Tutor3 and the next section describes the adaptive hypertext author's view. Then Section 5 describes the evaluation of Tutor's support for user control with the results in Section 6. We conclude with a discussion of lessons learnt from the current series of experiments with Tutor.

Background

There is a growing debate within the field of adaptive systems over how much control should be given to the user as well as how much transparency there should be to the inner workings of the system. A debate on direct manipulation interfaces vs. interfaces driven by software agents highlighted understanding and control as key usability issues with adaptive systems (Maes, Schneiderman, 1997). An agreed outcome of that discussion was that users need to understand the adaptive operation of an adaptive system in order to trust it to perform tasks on their behalf. In addition, users must feel as though they have ultimate control over the system if and when they choose to exercise it. We believe that understandability of an adaptive system's response and empowerment of the user to control this response are key usability issues that are yet to be fully addressed by designers of adaptive systems. This is vital if we are to trust adaptive systems to perform critical tasks on our behalf. We extrapolate this to non-critical tasks.

Another motivation for helping the user understand the adaptive operation is that increased privacy laws in many countries stipulate that where a system stores personal information about a user, the user has a right to view and to modify that information and understand how it is used (Kobsa, 2002). Ultimately, this requires that the user not only has access to their user model but also understands, to some degree, how the adaptive system processes their user model to produce an adaptive response.

Another possible reason for scrutability relates to the possibility that the adaptive hypertext could have errors in the adaptation. The more complicated the hypertext, the less efficient the authoring (Calvi, Cristea, 2002) and the more opportunities exist for authoring errors. One approach to this problem is to improve authoring tools. This is undoubtedly an important field of research. For example, tools have been developed for AHA! (Calvi, Cristea, 2002, Cristea, Aroyo 2002).

While AHA! has a similar architecture and adaptivity features to Tutor, its authoring tools include a specialised editor to define concepts and adaptation rules and a Graph editor which allows the author to visually link concept nodes (De Bra et.al. 2003, Calvi, Cristea, 2002, Cristea, Aroyo 2002). Such authoring tools will help reduce the difficulty in authoring adaptive hypertext and also reduce the number of errors. However, it is not possible to completely eliminate mistakes. In fact, a scrutable interface could be used as a debugging tool by the adaptive hypertext author as it should clearly explain how a page was adapted to the user.

A closely related problem can occur where there is mismatch between the user's understanding and the author's intention. For example, an adaptive maths course may offer the user the choice of an easy or hard version of the course. A user might request the easy version, only to find it is too easy and that they would have actually preferred more challenging material.

There is a substantial and growing body of work that makes the user model available to the user (Bull S, P Brna, and H Pain, 1995). In particular, in educational contexts, there has been considerable interest in the potential of supporting reflective learning by creating suitable interfaces to the user model as reported, for example, in two workshops (Bull, Brna, Dimitrova, 2003; Morales, Pain, Bull and Kay, 1999). Even in systems where the work is not explicitly concerned with supporting student reflection, there are many cases where the learner is given some insight into the system's model of the learner. For example, Corbett and Anderson (1995) describe a teaching system that provides a set of 'skillometers' which show a summary of the learner model in terms of the degree to which a set of skills has been learnt. Of course, displaying the user model provides the user with access to some understanding of just one of the core elements of the personalisation.

There has also been some work on providing greater user control in adaptive hypertext. For example, a website for a conference was used for an empirical

study to explore user's reactions to controllability of the adaptive system (Jameson, Schwarzkopf, 2002). The system was intended to help attendees compile a program of events they would attend at the conference and made personalised recommendations of events that might interest the user, based on their interaction with the system. Usage and subjective feedback was compared for different configurations of the system: (1) the system made recommendations only when the user requested them (2) the system made recommendations automatically. Users played with the system in all configurations and provided subjective feedback. The study found that both configurations had their advantages in certain situations and based on the user's personal preferences. However, the type of user control we are exploring is quite different to that of this study. In the first system configuration, the user could basically invoke the adaptive function when desired. However, the user could not, for example, control the outcome of this function. Nor could the user delve into their user model and update it to directly change the type of recommendations provided by the system. What if, for example, the system believed the user was interested in attending presentations about intelligent agents when the user was actually interested in direct manipulation interfaces? It is this type of control that we are exploring with Tutor.

POP (Höök, Karlgren, Waern, Dahlbäck, Jansson, Karlgren, and Lemaire, 1996) represents another important piece of work in this it area. It was an adaptive hypertext help system which aimed to achieve transparency and user control. It did this by allowing users to drive the user modeling by specifying their task, expressed in terms they were familiar with. It supported exploration of the adaptive hypertext with specially marked hotwords, which the user could select to see a set of questions they could ask about unfamiliar concepts. The adaptive content could be opened or closed. This represents an interface with support for user control of the user model and the ability to see the full hypertext. It did not enable the user to see why a particular part of the text was made directly visible, or not, because of the details of the user model. Notably, however, POP had a quite complex internal structure which was hidden in its internal black box. This meant it was then impossible for the user to go beyond viewing the user model.

An important aspect of our work on Tutor has been a focus on simplicity. To place our work in context, the adaptivity features and architecture of Tutor are similar to the early versions of AHA! (DeBra et. al. 2003). However, we have focused on building a user interface to facilitate scrutability rather than extending the adaptive hypertext capabilities.

In summary, several researchers have indicated the importance of user control, especially in learning environments. Then, in addition, several researchers have explored ways to make the user model open to learners and to use this to support learning. However, Tutor represents the first system to go beyond open, or even scrutable user models; it makes the *adaptivity* and *associated processes* open to the learner and controllable. The systems mentioned above have enabled the user to access, view and perhaps modify the user model, but not the

process driving the adaptation; it si this process which has been highlighted as equally as important by Maes and Schneiderman (1997).

Essentially, our goal of building a scrutably adaptive hypertext creates two quite different classes of challenge. On the one hand, we need to design an architecture for an adaptive hypertext system and to do this in a way that makes it possible for the user to scrutinise the adaptation processes. This poses considerable technical challenges. The other, quite different class of challenge is the creation of a suitable interface that supports users in scrutinising the adaptivity. Clearly, there is a real interaction between these two aspects: we would expect that it is far more difficult to build effective interfaces to more complex systems, with highly sophisticated representations for the user model as well as complex mechanisms for representing and generating the hypertext. For example, if the generation is based upon deep natural language generation with complex planning processes and the user model is an executable representation, it is not at all obvious how to make all these aspects and their interactions accessible and understandable to a user. Accordingly, we decided to begin with a very modest form of adaptation and to evaluate that before proceeding to more sophisticated forms. We were also encouraged in this decision since it seems that quite modest levels of adaptation, with correspondingly simple user models, may be quite useful. For example, Strachan, Andserson, Sneesby and Evans (2000) observed that users liked customisations based on a simple user model.

The design of the Tutor interface has been strongly influenced by the ELM-ART system (Brusilovsky, Schwarz, Weber, 1996, Brusilovsky, Weber, 2001), which teaches the LISP computer language. As indicated in the following description, the elements of the interface, including the course map and uses of colour, follow what appears to have been a successful interface for that system.

Increasing user control over adaptivity raises a new set of challenges. Given full control of their user model and the adaptivity, the user could sabotage the function of system by changing their user model beliefs such that the model no longer reflects them accurately (Kay, 2001). The user could quite easily undo all the hard work the adaptive system has done to form its assessment of the user. One remedy for this is to restrict access to parts the user model. A different approach, used in Mr. Collins (Bull, Pain, 1995), is to force the user to negotiate changes to their user model. Mr. Collins allows the user to inspect their user model to see beliefs the system holds about them, offer their own self assessment and then negotiate with the system to change its beliefs about the user. The user has to correctly answer test questions to convince the system.

Another problem of added user control and transparency is the additional cognitive load that it places on the student. The user of an adaptive system is busy using that system for its intended purpose, whether it is learning, searching, entertainment or something else. We envisage that typically the user's interactions with the scrutability component will be brief and infrequent. Some likely scenarios are:

- the user may become curious about the adaptivity;
- the user might wonder what content they are missing out on;
- a page presented to the user seems to contain irrelevant content;
- the user believes their preferences, interests or knowledge have changed or are not being catered to by the system;
- the user is simply experimenting with the interface.

Despite our argument for the importance of use control and transparency, we do not expect user to frequently exercise their control. This contrasts with the case studied by Jameson and Schwarzkopf (2002). Indeed, we observed that users only scrutinised the adaptation infrequently in out field trials of Tutor (Czarkowski and Kay, 2000).

However, when the user chooses to take control, this should be possible. In fact, this highlights the difficulty in developing an interface to support scrutability. Not only are adaptivity and scrutability relatively new concepts for many users, the interface must be simple enough for a casual user to understand yet powerful enough to support the expert user.

It is possible that scrutability may need to become a more frequently used feature of an adaptive system, if humans are to trust adaptive systems to perform more critical tasks on their behalf. Naturally, in cases where users employ adaptive systems or software agents to carry out important work or make decisions on their behalf, they would expect to have the ability to understand, and as needed to change, the systems behaviour and the process with which it makes decisions.

Tutor, has gone through several versions. The first version was evaluated in a field trial (Czarkowski and Kay, 2000). This appeared to be quite successful, with a total of 113 students registering with the system, 29% exploring the adaptivity and, of these, 27% checking what had been included as part of the adaptation and where content had been excluded. This evaluation identified limitations, too. While it did show material that had been adaptively included, it showed only the location of material excluded. Also, some users failed to appreciate that they could affect the adaptation, at any time, by altering their answers to the profile questions. In essence, the Tutor evaluation showed promise but also identified ways that the scrutability support was incomplete. These concerns were addressed in Tutor2. We performed a qualitative evaluation to assess whether users could scrutinise its adaptation effectively (Czarkowski, Kay, 2002, 2003). This evaluation indicated that users could *not* do the basic scrutability tasks we had defined as essential. So, in light of these findings, we completely redesigned the adaptation explanation to the form it has in Tutor3, the version described in this paper.

Overview of Tutor from the student perspective

To use Tutor the student must first register for a username and password. Once registered, they log in and select from the list of available courses. On entering a course, Tutor requires the user to complete a questionnaire form (see Figure 3). The student's answers define their user model for the course. Students

entering the course for the first time will have a blank profile and must fill in answers on the questionnaire. Those who have previously accessed the course have the option of updating their previous answers. When the student saves their profile, Tutor navigates to the teacher's instructions page.

The teacher's instructions page, like all lesson pages, has the icons that can be seen in Figures 1 and 2. Each has its text description as a mouse over. These icons are, in the order that they appear on the screenshots:

- course map – gives an overview of the whole course and the student's progress through the course;
- teacher's instructions page – which is the first page the student sees after the login and profile validation process and so this is the place for the teacher's announcements and bulletin board. It is used as the course home page but has all the features of a regular lesson page;
- notes editor - a free-form set of arbitrary entries by the student, supporting the student in actively recording information as they work through the course;
- glossary – essentially another adaptable page, but generally would have many links to it through the course;
- profile page – shows the questionnaire used to establish the user model and can be used at any time to update it;
- exit – to leave the course;
- on-line help page – at the far right; this describes the interface and system features, including adaptation and scrutability support.

As we have already mentioned, Figure 1 and Figure 2 show how a typical lesson page is adapted differently for students who have different user models. The page in Figure 1 has been adapted to a student who is revising the course and wants to gain a mastery of all the course material. In contrast, the page in Figure 2 has been adapted to a student who is hoping to learn just enough to pass. Note that the page in Figure 1 has an additional (fourth) paragraph of text and a table that lists conventionally used UNIX directory names. The hypertext author considered this material was not needed to gain a bare pass. So, when the user model indicates that this is all the user wants to do, it is not shown. However, where the user model indicates the user wants a broader understanding, it is included. Note also that in Figure 1 an additional right arrow labelled *6.0 Review Questions* has been included in the page. This is a hyperlink to another lesson page that contains additional review questions. It appears only where the user model indicates the user wants to revise the course material.

Profile Page

The profile page presents a set of questions and uses their answers to establish the user model. An example of a typical profile page is shown in Figure 3.

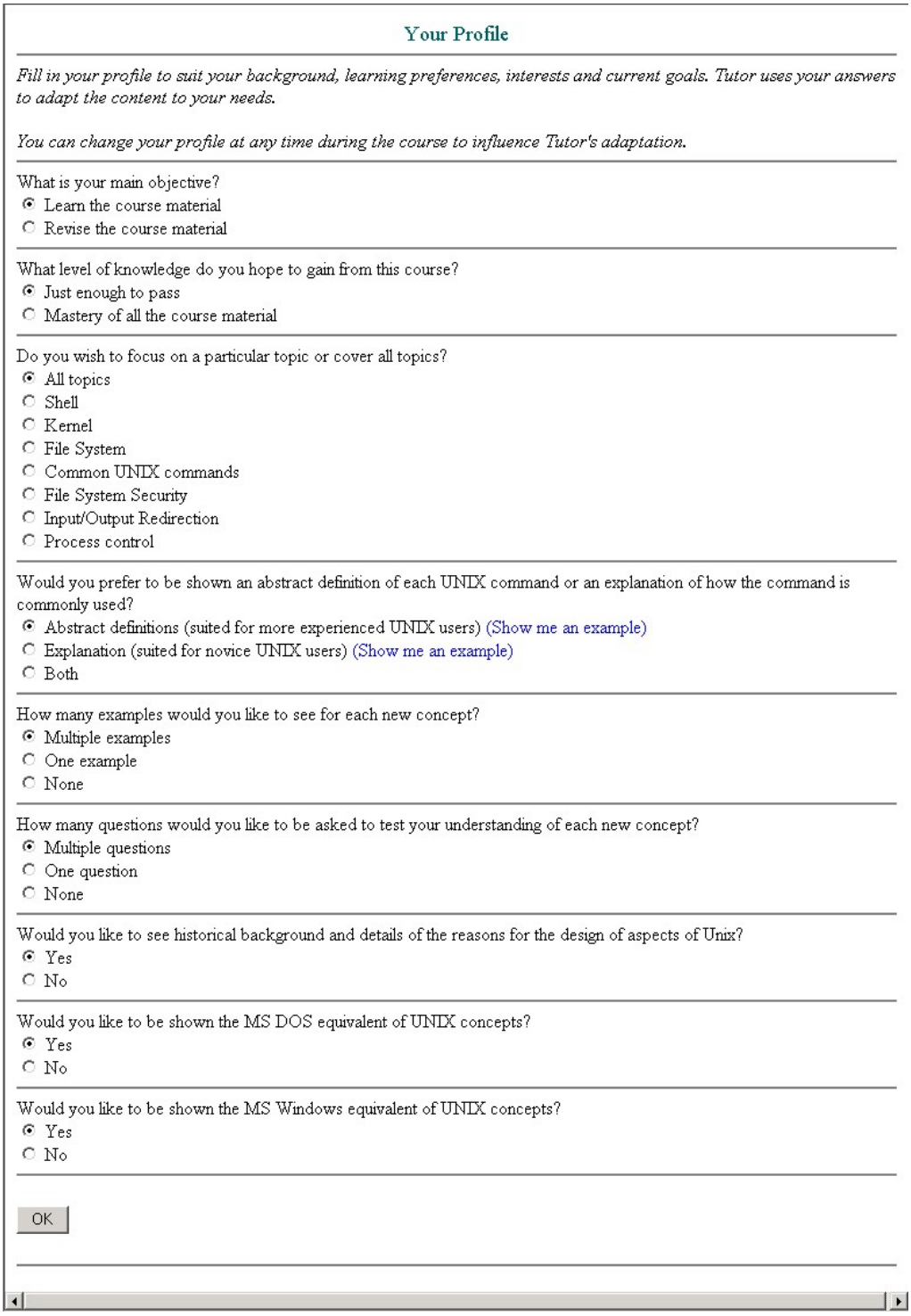


Figure 3. Example of a profile page.

The student must complete every question on the profile page before they are shown course material. Every time a student enters the course, Tutor displays the profile page pre-populated with the student's current data. This gives the student an opportunity to reflect on their learning strategy, immediate goals and interests and to update their profile should they wish to do so.

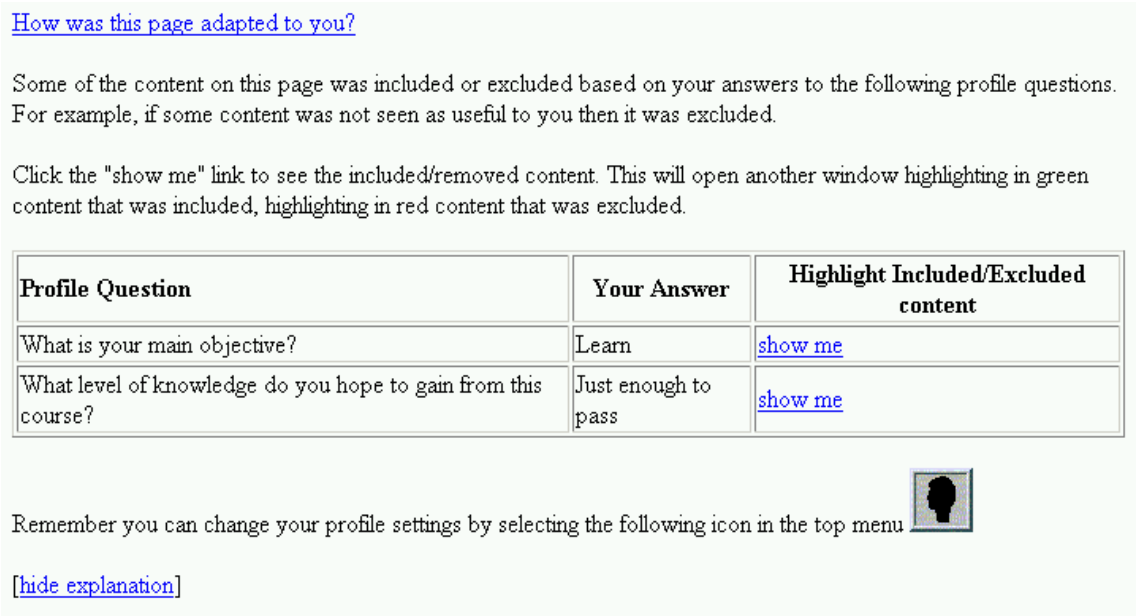
Tutor provides the possibility of links to examples of the effect of a question. This is intended for cases where it is hard to formulate the question clearly and where an example of its effect should be clearer. The sample page can be accessed by the student via a hyperlink labelled *Show me an example* positioned next to the profile option. For example, for the fourth profile question in Figure 3, the user can view a sample of a typical lesson page that has abstract definitions of concepts, and another version of the same page showing what the author has called explanations.

The profile page is important as it central to the scrutability component of Tutor. The student must understand how to read and edit their profile in order to change the personalisation to achieve the desired effect. Note that there is explanatory information at the top of on the profile.

Scrutinising the adaptation

On pages where content has been adaptively included or excluded, Tutor dynamically includes the hyperlink *How was this page adapted to you?* at the bottom of the page. On pages with no adaptation (as in Figure 9), the text "There was no adaptation on this page appears instead of the link.

Clicking this link displays an adaptation summary, just below the link. For example, the adaptation summary that would be displayed for Figure 2 is shown in Figure 4. This summary is the starting point for scrutinising the adaptation of the current page. It tells the student which elements from their profile caused the adaptation and allows the student to probe even deeper, to see specifically what has been included or excluded because of their profile.


How was this page adapted to you?

Some of the content on this page was included or excluded based on your answers to the following profile questions. For example, if some content was not seen as useful to you then it was excluded.

Click the "show me" link to see the included/removed content. This will open another window highlighting in green content that was included, highlighting in red content that was excluded.

Profile Question	Your Answer	Highlight Included/Excluded content
What is your main objective?	Learn	show me
What level of knowledge do you hope to gain from this course?	Just enough to pass	show me

Remember you can change your profile settings by selecting the following icon in the top menu

[hide explanation]

Figure 4. Example of an Adaptation Summary on a page that has been adapted to a student.

The adaptation summary has several elements. It begins with information describing how to scrutinise the adaptation. Also, the text at the bottom of the adaptation summary reminds the user that they can change their profile settings by clicking the user profile icon in the top menu.

The core of this page is the list of each user model attribute that caused adaptation of the page. This is expressed in terms of the profile questions so that it should be familiar. Against each question is the user's current answer. The third column displays a hyperlink labelled *show me*. For example, the last row in Figure 4 indicates that there has been adaptation because the student answered *Just enough to pass* for the profile question *What level of knowledge do you hope to gain from this course?*

Clicking this hyperlink opens a separate browser window to display the Adaptation Explanation page in Figure 5 and this indicates adapted content by the background colour. It shows adaptation associated with the single profile question that the user selected with the *show me* link. It is in a separate window so that the user can compare this annotated version of the page to the original version.

The colour choices follow a traffic light metaphor, an idea inspired by ELM-ART (Brusilovsky, Schwarz, Weber, 1996, Brusilovsky, Weber, 2001): content that was included is highlighted with a green background (which is the light grey in the figure). Content that was excluded is highlighted in red (which appears as a darker grey in the picture). This is explained to the user in the key at the top of the adaptation explanation page. The key shows the selected profile question, the students current answer and explains the colour coding.

For example, from Figure 4, if the user clicks the *show me* hyperlink next to the profile question *What level of knowledge do you hope to gain from this course?*, the adaptation explanation page will be displayed as in Figure 5. On this page, the fourth paragraph under the section titled *1.3 Directories* and the table below it are highlighted in red, indicating the content was adaptively excluded from the page. In our test environment with Netscape 4.7 browsers, moving the mouse over the *i* icon (just above the red paragraph) pops up the following text informing the user why the context was excluded, and how to change this adaptation:

This content was excluded since your answer to the above profile question was 'Just enough to pass'. To make this content included, change your answer to 'Mastery of all the course material' in the profile editor by clicking the head icon in the main window.

In Internet Explorer 5+ browsers the text is displayed by default underneath the adaptive content, as it appears in Figure 5.

From the adaptation summary table on the main page (Figure 4), the student can open a separate adaptation explanation page for each profile attribute that

affected adaptation and thus compare how different profile questions affected the adaptation of the page.

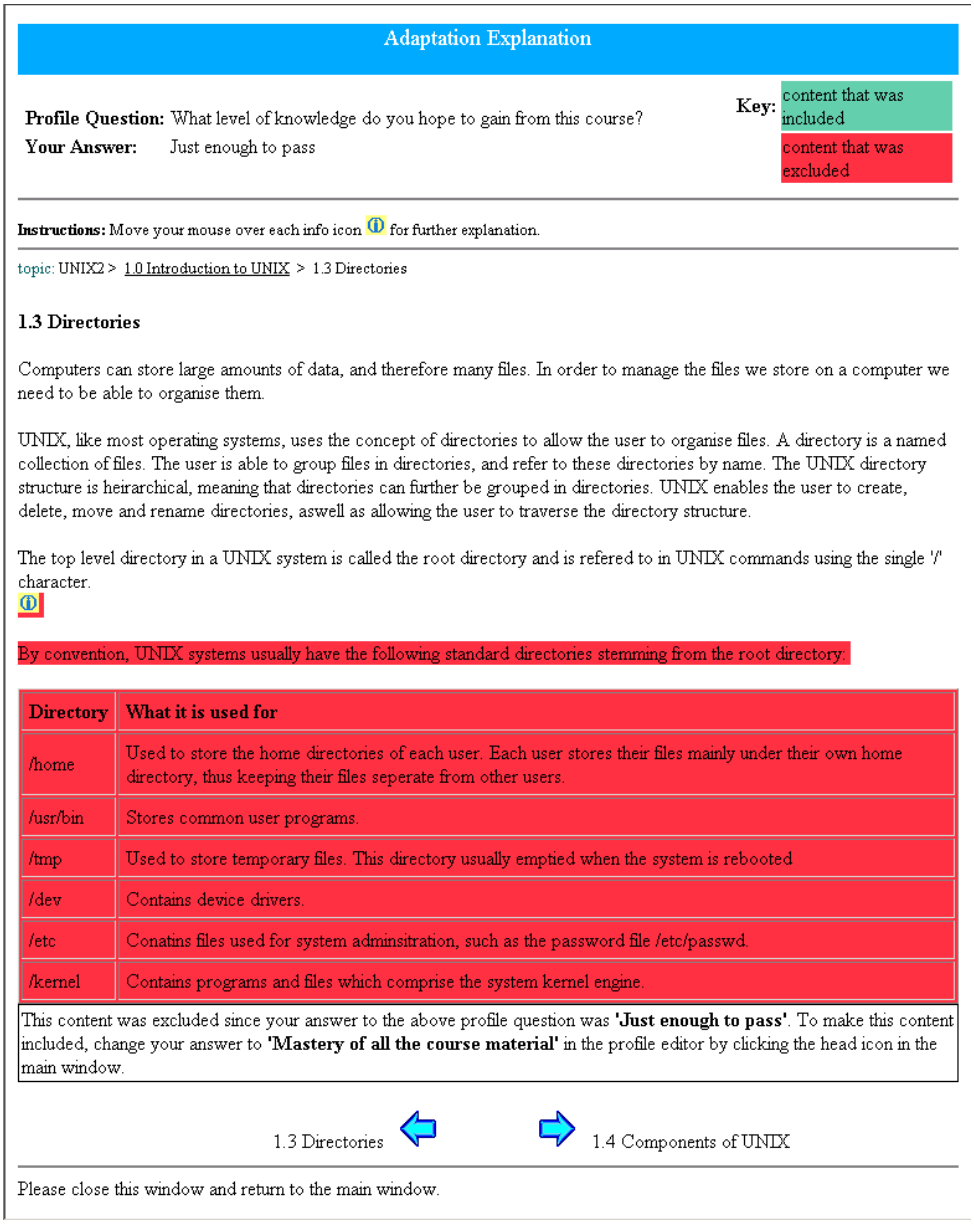


Figure 5. Example of an Adaptation Explanation page (as seen in Internet Explorer) – accessed by clicking the *show me* link in Figure 4 for the profile question *What level of knowledge do you hope to gain from this course?*

Course map page

The course map, such as the one shown in Figure 6, provides an additional navigation option that allows students to randomly access course pages. This complements the links that authors will typically create on each page of the course. The course map displays a list of hyperlinks to all lesson pages in a course. Hyperlinks are cascaded to indicate the hierarchical relationship between pages. For example, the first page of a new chapter is left justified, with pages within that chapter being offset to the right.

Whilst the list of hyperlinks to pages is the same for each user, the hyperlinks are colour coded to provide adaptive navigational guidance. As in ELM-ART (Brusilovsky, Schwarz, Weber, 1996, Brusilovsky, Weber, 2001), colour coding of hyperlinks in Tutor (both in the course map and on a lesson page) is based on a traffic light metaphor. For any hyperlink, the author may specify zero or more pre-requisite pages. If the student has visited all the pre-requisite pages, the hyperlink will appears in green, indicating the student is ready to learn the material. If the student has not visited all the pre-requisite pages, the hyperlink appears in red. If the student has already visited a page, its hyperlink will appear in black, but still underlined to distinguish it from normal text. Hyperlink colour coding indicates which pages the hypertext author considers the student should read or not, based on the current state of the user's user model. At the same time, the student is free to access any page.

Once opened, the course map stays open until closed. One way to use Tutor is to keep the course map open while working through the course. Clicking a hyperlink in the course map loads the lesson page in the main browser window. If the course map is left opened, its display is updated as the user accesses a lesson page, updating the effect of pre-requisites that become satisfied. For example, Figure 6 shows a course map where the student has accessed the first page of the course. The hyperlink to the first page, *1.0 Introduction to UNIX*, appears in black. Since the first page was a pre-requisite for the second and third pages, their hyperlinks are green (in Figure 6 *History of UNIX* and *1.2 Files* are green). All other pages in the course map have the second and third pages as pre-requisites, hence they are all displayed in red.

Notes editor

The notes editor enables the student to make their own notes while they are working through a course. Each time the notes editor is accessed the course name and current timestamp are appended to the current notes. See Figure 7. The student's notes are stored by Tutor and displayed when the notes editor is accessed.

Clicking the *Save and Print* button parses the notes (which can be marked-up in HTML as shown in the figure) and renders this as a printable page that the student can print or save to their local machine.

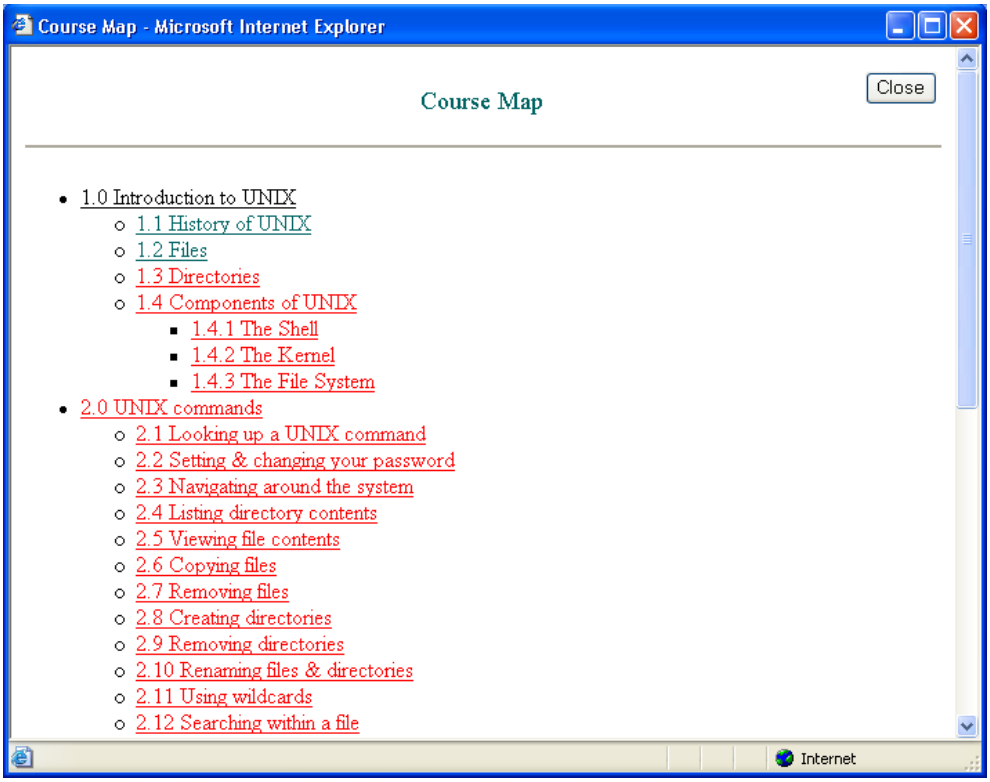


Figure 6. Example of the Course map.

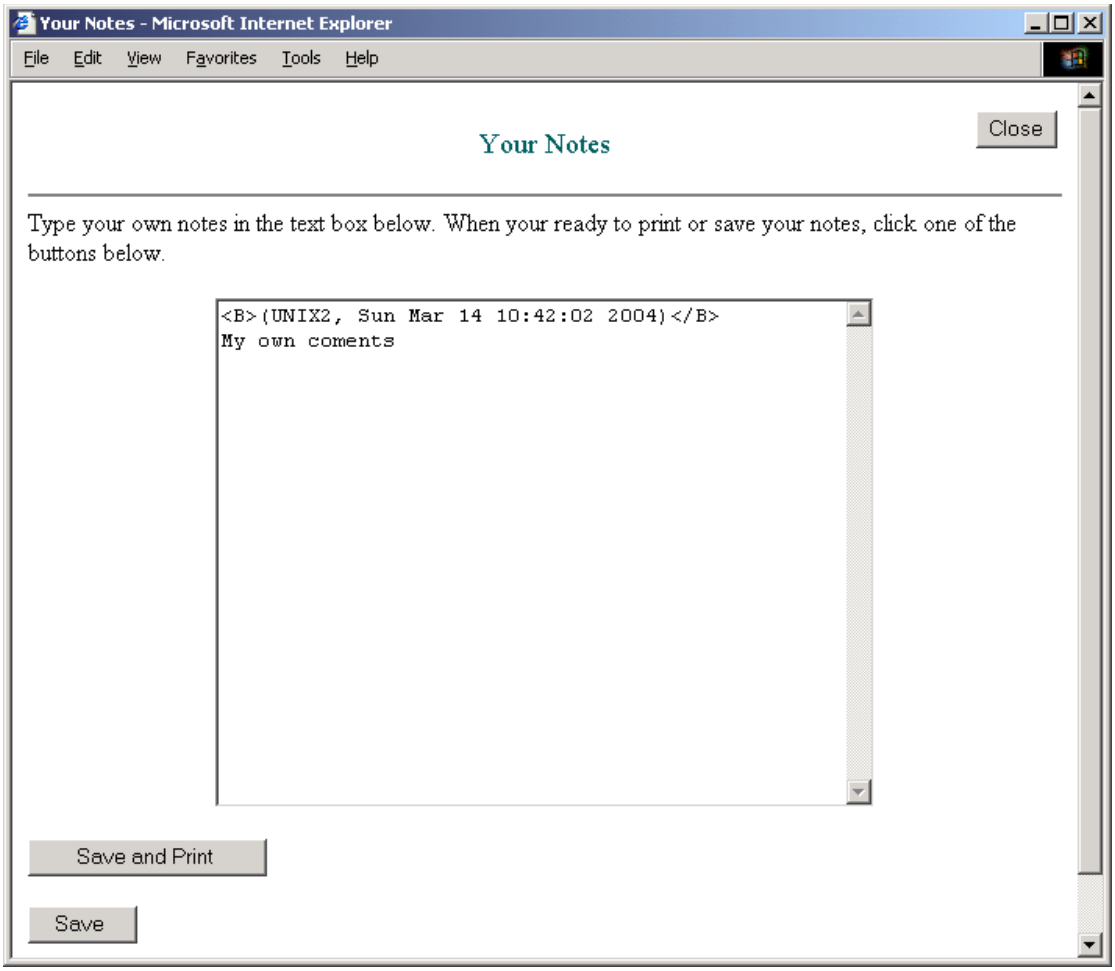


Figure 7. Notes editor showing notes in raw form as entered by the student.

Glossary page

This is intended for the usually glossary role, with descriptions of terms used in the course. An example is shown in Figure 9. The glossary page can be adapted and that adaptivity scrutinised as for other pages, however, Figure 9 shows a non –adaptive page to raise the point Tutor3 also inform the user when there was no adaptation on the page. Glossary pages that are adaptive include the scrutability interface as in Figure 4. Figure 8 shows the annotation of a hyperlink to the glossary entry for the word *group*. The glossary word is displayed in bold face and the hyperlink is annotated with the text *see glossary*, displayed in green font. Hyperlinks to glossary entries are annotated in this way, that is, differently from regular hyperlinks, since they operate differently. Clicking the glossary hyperlink in Figure 8 opens the glossary page in a separate browser window and jumps to the selected word as shown in Figure 9 whereas clicking a regular hyperlink navigates to the anchored page, losing the information on the current page. The glossary hyperlink allows the student to continue with the main task in the lesson window, whilst consulting the glossary. Once the user understands this operation, the intention is that they will feel comfortable that they can click this link and view the glossary definition without loosing their current place in the main lesson page.

one set for the **group** (see glossary) to which the owner belongs - specifies the type of access every user in the group may perform

Figure 8. Annotation of hyperlinks to glossary entries.

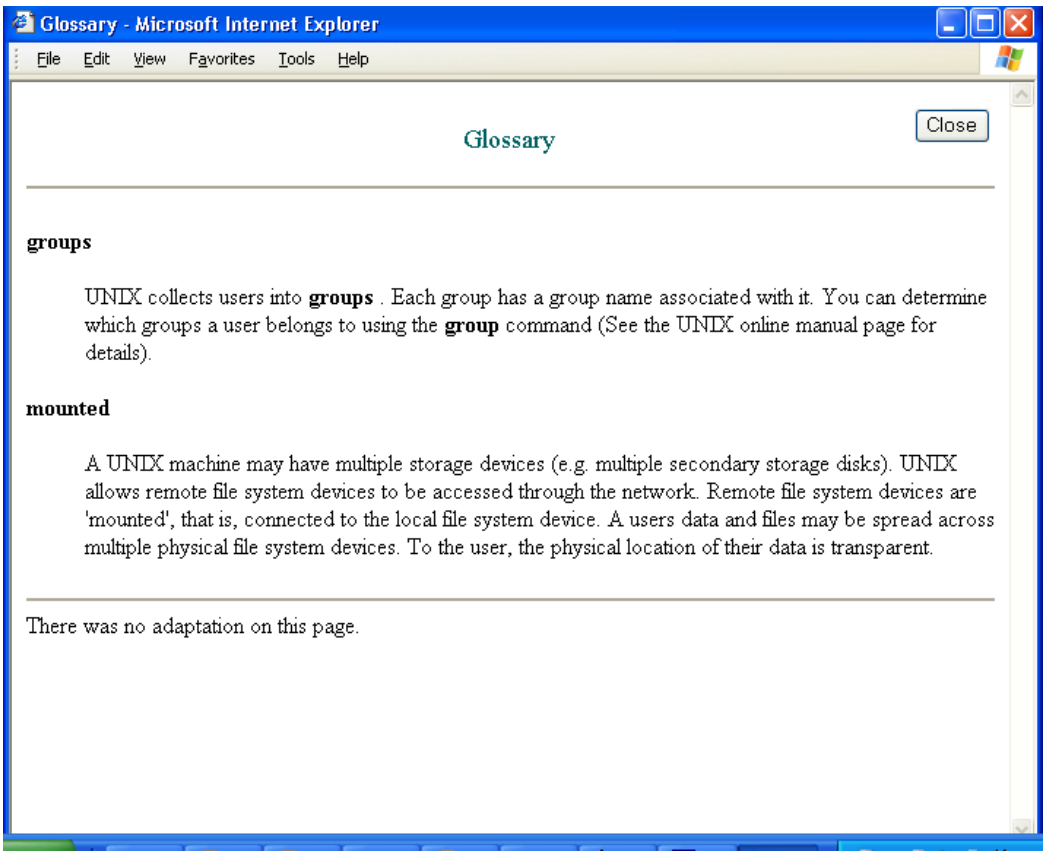


Figure 9. Typical Glossary page. Lesson and glossary pages that do not have adaptive content display the text There was no adaptation on this page at the bottom of the page. In a previous version of Tutor we displayed the link *How was this page adapted to you?* whether there was adaptation or not and we found if students clicked on the link to reveal no adaptation, they were very unlikely to try the link on another page.

Overview of Tutor for the adaptive hypertext author

Tutor's adaptivity is driven by the markup of any HTML document with its ATML language. We have designed ATML to make it easier for an author to create adaptive hypertext. ATML has mark-up elements for interactive multiple choice questions, chapter and topic headings, glossary hyperlinks and page hyperlinks. If the author includes these tags in the lesson, Tutor dynamically translates this into HTML and generates the JavaScript code for the interactive features. This means that the author can ignore style and presentation of the course material and is free to concentrate on the teaching content.

ATML conforms to the XML 1.0 specification, and allows the user to include HTML 4.0 elements which are XML compliant. To create a new course in Tutor, the author first creates a set of ATML documents:

- um.xml - to generate the profile page.
- coursemap.xml - to generate the course map page.
- teacher.xml –to generate the teacher's instructions page, which the teacher can use as a bulletin board.
- glossary.xml –to generate the glossary page.
- <pagename>.xml –adaptive lesson pages of teaching material.

The author can also add elements to a style sheet (CSS1 compliant) definition document. The styles can then be used in the course material. Once these documents are uploaded to the server web site, the new course is ready to be accessed by students. The following sections will describe ATML more detail.

ATML - Profile page

An extract of the profile page ATML used to generate the page in Figure 3 is shown below. The *course* element defines a globally unique course identifier, in this case *UNIX2*. This tells Tutor that the following user model attributes relate to the UNIX2 course. Next the document contains definitions for each user model attribute (*pref* element), the questionnaire text that is displayed on the profile page (*desc* element) and the possible answers (*answer* elements).

```
<?xml version="1.0"?>
<um>
<course id="UNIX2" />

<pref id="MainGoal">
    <desc text="What is your main objective?" />
    <answer value="Learn">Learn the course material</answer>
    <answer value="Revise">Revise the course material</answer>
</pref>

<pref id="Level">
    <desc text="What level of knowledge do you hope to gain from this
course?" />
    <answer value="Just enough to pass">Just enough to pass</answer>
    <answer value="Mastery of all the course material">Mastery of all the
course material</answer>
</pref>

...
```

```
</um>
```

Figure 10. Excerpt of ATML used to generate profile page in Figure 3.

ATML – Adaptive lesson page

An ATML lesson page may contain adaptive and non-adaptive teaching material written in HTML or sourced from external HTML documents, or ATML tags to include interactive multiple-choice questions and various ATML link types. An excerpt of the ATML used to generate the lesson page in Figure 1 is shown in Figure 11.

```
<?xml version="1.0"?>
<lesson>
<page>

<topic><link src="Courses/UNIX2/p1.xml" text="1.0 Introduction to UNIX"/>
</topic>
<title>1.3 Directories</title>

<P/>Computers can store large amounts of data, and therefore many files. In
order to
manage the files we store on a computer we need to be able to organise them.

...

<adapt cond="Level" value="Mastery of all the course material">
By convention, UNIX systems usually have the following standard directories
stemming from the root directory:
</adapt>

...

<adapt cond="MainGoal" value="Revise">
<pagelink>
<prev auto="true" />
<next src="Courses/UNIX2/p6.xml" title="6.0 Review Questions"/>
</pagelink>
</adapt>

</page>

</lesson>
```

Figure 11. Excerpt of ATML used to generate lesson page in Figure 1.

Of particular importance is the *adapt* tag since it controls adaptation. In Figure 11, the paragraph starting with "By Convention …" will only be displayed to users whose user model shows a value of *Mastery of all the course material* for the attribute *Level*. Referring to the profile page ATML, Figure 3, it can be seen that this attribute is the profile question *What level of knowledge do you hope to gain from this course?*. The low level attribute name *Level* is used internally and is not displayed to the user. This approach was chosen to avoid verbose ATML. Figure 11 also shows how ATML can be used to generate adaptive page links. In this example, a link to the page titled *6.0 Review Questions* is included to users who have indicated revision as their main objective. The rendered page link can be seen in Figure 1. A similar approach is used by the AHA system

(DeBra et. al., 2001). It also uses mark-up to describe adaptive fragments that are conditionally displayed to the user if the user model satisfies the inclusion criteria. The mark-up used by AHA is more expressive than Tutor, but hence more complicated.

Evaluation of Tutor support for scrutability

At a high level, the aim of our experiment was to gain insight to whether our tool for scruatability was effective. For it to be effective meant the users

- could understand the purpose of profile questions was to establish their user model;
- were able to determine what was adapted;
- could demonstrate control over adaptation by changing answers to profile questions.

Our evaluation of Tutor3 was qualitative, based on a think-aloud (Neilsen, 1994). This has the merit of being relatively low cost and giving insights into the causes of difficulties.

Selection of Participants

Following Neilsen, we selected five participants for the evaluation. They have different backgrounds and varying degrees of computer literacy: one was a secondary school student, two were third year computer science degree students and there were two adult participants with basic computer literacy skills. Table 1 shows a summary of each participant's age group, education level and computer literacy.

Table 1. Summary of participant backgrounds.

	Participant ID				
	1	**2**	**3**	**4**	**5**
Age group	21-25	21-25	18-21	18-21	14-18
Education Level	Post Secondary school study	Post Tertiary study	Computing undergrad	Computing undergrad	Secondary school
Computer literacy	Basic	Basic	High	High	Moderate

Overall, they represent a quite computer literate group; all had previously used the internet. Due to the small number of participants we can not make claims about the implications for all world wide web users and we have to interpret findings with a some caution.

The Participant's Task

We designed a scenario around a fictitious person, Fred. This meant that all participants were dealing with the same student profile and we could predict the exact adaptation each should see. This, in turn, meant that all participants did exactly the same task and should have seen exactly the same screens.

Each participant was provided with a worksheet. This described Fred's learning goals, interests and background. Participants were asked to assume the role and background of Fred and use Tutor3 to start working through the beginning of the Introductory UNIX course. Participants were presented with one page of the worksheet at a time so they could not jump ahead. Participants we allowed to spend as much time as required. We observed the participants as they completed the task to record their comments. In addition, the system logged all their interactions with the interface. The participants were not given any training or a demonstration of the system.

The tasks of the worksheet were designed so that each basic issue was explored in three subtasks. This provided internal consistency checks, an important concern since there are degrees of understanding and we wanted insight into just how well each participant was able to scrutinise the adaptation. Extracted questions from the worksheet are shown in Table 2. These reflect our core evaluation goals as described at the beginning of this section.

Table 2. A sample of questions in a worksheet completed by participants in the evaluation of Tutor3.

Part	Concept	Questions presented in worksheet
1	Understanding the purpose of profile questions	• Will Tutor use Fred's profile settings? If so, what will it use Fred's profile settings for? • Where does Tutor get information about Fred to use to perform adaptation of the content to suit him? • Will Fred be able to influence or control the way Tutor adapts content to him? If so how?
• 2	Ability to determine what was adapted	• What would you do in the system to find out whether Tutor adapted any material on the Teacher's Instructions page to you? • Did your answer to the profile question *What is your main objective?* have any effect on the contents of the Teacher's Instructions page? How do you know this? • Was any content specifically excluded because of your answer to the profile question *What is your main objective?* If so, what was the content?
• 3	Demonstrating control over adaptation by changing answers to profile questions	• Now consider what the first sentence on the page would read had you answered the profile question *What is your main objective* with *Revise the material*. Without changing your answer just yet, write out how you think the sentence would read. *By the end of this course you will have …* • Explain what actions you would have to perform in the system to change your answer to the profile question *What is your main objective?* • Change your answer for the profile question *What is your main objective* to *Revise the material*.

The first part of the worksheet instructed the user to register with the system, log on and select the Introductory UNIX course. The first task was to complete the profile of Fred by answering questions on the profile page (as in Figure 3). The worksheet did not specify the answers for each question but did describe Fred's learning goals, interests and background. Each participant was then asked questions from the first block in Table 2.

The second part of the worksheet asked each participant to navigate to a specific page. Then, participants were asked to determine whether any of the content on that page was adapted to Fred, based on the answer to the profile question *What is your main objective?* The participants had to indicate any adapted content and whether it was included or excluded. To perform this task, the user had to notice and click the hyperlink *How was this page adapted to you?*

The third part of the worksheet involved what-if experiments. The participant was asked to guess the content of a specific paragraph, assuming Fred were to change his answer for the profile question *What is your main objective?* to *Revise the material*. Answering this question should not have required any guesswork. The answer is on Adaptation Explanation as in Figure 5.

Results

Understanding the purpose of profile questions

The participants first had to answer the 8 profile questions, as Fred would have. These appeared as in Figure 3, with additional questions about preferences for abstract explanations, numbers of examples and self-test questions, interest in historical background and understanding of other operating systems. For this experiment, the critical aspects about Fred's background were that he wanted to learn (not revise) and he was not interested in historical background information. All participants answered these profiles questions consistently with the information we provided about Fred which meant that the pages displayed to them in the experiment had the same adaptations.

The participants completed the profile and answered the first set of questions from Table 2. At this stage, they have not been shown any adapted content and have not had any opportunity to play with the scrutable interface. Their answers to the worksheet were based purely on the information shown in the profile page and any prior experience with adaptive systems. Note that text on the profile page, included below, already gives away the answers.

"Fill in your profile to suit your background, learning preferences, interests and current goals. Tutor uses your answers to adapt the content to your needs. You can change your profile at any time during the course to influence Tutor's adaptation."

In their answers to the first two questions, all participants stated their profile answers would influence the way material would be presented to them. However, participants had difficulty with the question *Will Fred be able to influence or control the way Tutor*

adapts content to him? If so how? Participants 1 and 2 correctly stated that by answering the profile questions in a way that suits Fred, they had control over the adaptation. Participants 4 and 5, on the other had, answered "No". Participant 3 answered "Yes, but I don't know how." Through questioning participants about their answers after the experiment, we know that what participants 1,2,4,5 meant was that although they believed they could initially influence the system through their profile answers, they expected that once they started working through the course they would no longer be able to influence the system or change their profile. All believed there were factors influencing adaptation other than the profile they would not be able to control. Note that this is despite the help text on the profile page, as in Figure 3. Participant 5 is an interesting case because later on in the experiment when he realised he could change his profile answers, he returned to change his answer to state *"Yes, by changing the way he responds to things."*

Ability to determine what was adapted

The next part of the worksheet asked participants the navigate to a specific page, explain how it was adapted and explain whether the profile questions had any impact on this adaptation. Recognising the link between the profile questions and the adapted content was central to being able to understand and control the adaptation. We expected participants would find this exercise straightforward. However, this was quite challenging for most participants.

Participants 1 and 2 could not work out how to determine what was adapted and needed help in accessing the explanation in order to continue the worksheet. To perform this task, the user had to notice and click the hyperlink *How was this page adapted to you?* at the bottom of the page (as in Figure 1). Participant 1 answered the worksheet correctly but without looking at the adaptation explanation. This participant's answer to the profile question *What is your main objective?* was *Learn*. Thus, when they saw the word *Learn* was highlighted on the screen, they assumed it was adapted because of that profile question without really analysing the adaptation explanation. Participant 2 could not understand the adaptation explanation, believing the highlighted content was content Tutor wanted to annotate as important rather than annotate as adapted content.

Participant 3 also could not work out how to access the adaptation explanation. However, their work-around method for determining adapted content was to go back to the profile page, change the profile answers, navigate back to the lesson page and study the content for changes. This participant correctly identified the included content but struggled to identify excluded content visually.

In contrast, Participants 4 and 5 quite easily found the link to access the adaptation explanation, found the correlation between the highlighted adapted content and their profile answers. We observed that Participant 5, in particular, very carefully examined all the text on the screen. Other users looked at the pages but seemed to ignore any help text offered by the system.

Demonstrating control over adaptation by changing answers to profile questions

The final part of the worksheet tested whether the participant was able to demonstrate control over adaptation. In fact, this is really re-testing the previous concepts since if a participant truly understood how to determine what content was adapted and that the adaptation was due to their profile, they would already understand they can control adaptation by changing their profile answers.

The worksheet asked participants to alter the adaptation on a particular page. The first question asked the participant to guess what the content of the page would be, assuming a specific profile setting, without actually changing the profile. To answer this question, we expected the participant would examine the adaptation explanation which shows how the content would be affected based on a change to the profile.

Participant 1 examined the adaptation explanation carefully and learnt how to read it. They correctly guessed the adapted content and were able to change their profile to prove they were correct. Participant 2, was still confused and could not work out how to access the profile page. Participant 3, who had not yet discovered the adaptation explanation, was not able to guess the expected page content. Participant 4 had previously demonstrated an understanding of the adaptation explanation but did not realise that the facility could be used to predict the content of the page in this case. Participant 5 had no problems correctly answering and testing their answers to the worksheet.

There was no time limit imposed on completing the worksheet. Participant 2, who performed the worst, took about twice as long as most of the other participants, who took between 20 to 31 minutes. Since this included the time to complete the worksheet and to think-aloud, it is not indicative of the time needed to explore the way material is adapted.

Overall, only Participants 1 and 5 seemed to fully understand how to interpret the adaptation explanations and were able to use the explanations to control the inclusion/exclusion of content to a desired effect. Participant 2 understood the profile played a role in the adaptation but could not understand the adaptation explanation nor work out how to change the adaptation. Participant 3 did not discover the adaptation explanation until late in the evaluation but then demonstrated understanding of it.

Discussion and Conclusions

There are a number of key findings and concerns raised through this experiment. With the small sample size in the qualitative experiment, we need to be cautious about any strong claims. However, we can identify some important and interesting outcomes that are important for our goal of supporting scrutability and control over adaptation.

The participants in our study were comfortable our simple user models used to capture information to achieve personalisation

Our users did understand the concept of a user model. They appreciated that the system would store information about them and in return provide personalised material.

The participants in our study understood that they had input to the personalisation process, but needed convincing that they could control it as well

Having filled in their initial profile, all participants could appreciate that their user model would influence the personalisation of material. However, some of our users needed time to work out that they could control the personalisation. They appeared to believe that once they filled in their profile, that would be the end of their input. It seems that our users, perhaps due to previous experiences, assumed they did not have any control over the adaptation. One user expressed surprise at the extent to which they could control the adaptation. Notably, users learnt they could control adaptation and they achieved this by reading the instructional text presented by the system; no intervention was required.

Although user need convincing they have control over the Tutor's adaptation, it deserves mention that we have had feedback from users requesting tractability and control. Participants in evaluations of earlier Tutor versions (Czarkowski and Kay, 2000 and 2001), stated they want to see all the information available for a page, that is, to see the content that was excluded by adaptation. It seems that there is a group of users who do not always trust the adaptation and do want to see what they have missed out on.

Building an interface for scrutability support is difficult.

A key concern, highlighted by the experiment, was that users had difficulty finding the scrutability tools when needed. Participants 1, 2 and 3 required help to find the link *How was this page adapted to you?* Participant 4 found this link without help. Due to a browser anomaly, Participant 5 was presented with the adaptation explanation without having to click the link. This last serendipitous case suggests that it might be helpful if the user is presented with this information at critical times such as the first page viewing. This reduces the need for the user to discover, ab initio, where to look for the information.

Although we had provided some instructional text explaining how to access the adaptation explanation and profile page, users who skimmed over the interface tended to ignore this. They then had difficulty completing the evaluation. Also, Tutor has online help, accessed via the large question mark icon on each lesson page as well as on a hyperlink near the link *How was this page adapted to you?* None of our users accessed the online help.

One of the challenges of supporting scrutability is that we would expect users to only want to explore this facility very irregularly. This is not an element for which users would be trained as part of the normal use of the interface. Typically, users would have no need for it over long periods of time and use of the interface. However, when there is a suitable trigger, such as unexpected behaviour, we want the user to be able to work out how to delve into the adaptation. So, for example, the learner who has asked for the broader view of the course may have a busy week and decide that they need to focus on the core needed to pass in that week. At that point, they need to be able to work out how

to effect that change. Similarly, if the user is surprised to see quiz questions when they indicated that they did not want them, they should be able to check to see how those questions were put there, be that because the hypertext author made a mistake in the adaptation tag or simply decided to present them to all users, regardless of their user model.

At the beginning of this chapter, we identified four core parts of an adaptive hypertext. At this stage, Tutor has the simplest form for each of these that we could devise for a practical adaptive hypertext. It operates on the basis of a very simple user model and a single source of user modelling information, namely the user's answers to profile questions. This makes the interpretation of the user model very simple. Tutor has a very simple adaptation mechanism, based on the limited power of the ATML adaptation language. However, we have found that this basic level of personalisation still poses substantial interface challenges.

Overall, it seems that the notion of adaptivity was familiar to our users. This is consistent with the growing use of adaptivity. However, the idea that they had could control the adaptation was new. We have made progress in building a more effective interface for scrutability in the three versions of Tutor. On the basis of the current evaluations, if we imagine that these users had been using Tutor in an authentic learning context and they had cause to wonder about the adaptivity, it is unclear whether they would have thought to try to scrutinise the adaptivity. Even if they had found the relevant links, it is seems that some would not have been able to work out, unaided, exactly what was adapted and how. Our next step is to further refine the interface and to evaluate it in a more authentic environment where the users are actively trying to learn the information presented. We propose to introduce a small number of errors in the adaptation in the hope that these will create the motivation to scrutinise the adaptivity. In terms of refining our interface, we will explore ways of informing the user they can scrutinise the interface without requiring them to search for a seemingly invisible link at the bottom of the page. One idea so far is to introduce the scrutable interface on the profile page by providing a demonstration. This would also help to present the connection between the profile answers and the user's control over adaptation.

There are many situations in which a user might wish to understand and control the adaptivity of an adaptive hypertext system. Supporting this is proving quite difficult. Providing the user with the ability to understand or at least trace an adaptive system's response and provide input to the system to alter its adaptivity function are key usability issues that are yet to be fully explored by adaptive systems. However, these are key features adaptive system must have if users are to trust the adaptivity on critical tasks performed on their behalf.

The evaluations we have reported for the current version of Tutor confirm our previous observations of the interface challenges of supporting scrutability of adaptation based on simple user models and modelling processes and simple adaptivity processes. We have much work ahead in further refining the

interface support for scrutability of hypertext adaptation as well as extending it to more complex hypertext adaptations.

References

Bloom, B S, (1984) The 2 sigma problem: the search for methods of group instruction as effective as one on one tutoring. *Educational Researcher*, 13, 4-16.

Browne, D., Totterdell, P., Norman, M., (1990) Adaptive user interfaces, Academic Press, San Diego, California.

Bull, S, P Brna and V Dimitrova (eds), (2003) *Proceedings of Learner Modelling for Reflection Workshop, Volume V of the AIED2003 Supplementary Proceedings*, 209-218.

Bull S, P Brna, and H Pain, (1995) Extending the scope of the student model. *User Modeling and User-Adapted Interaction*, 5(1):45--65.

Bull, S. and Pain, H. (1995), Did I say what I think I said, and do you agree with me?: inspecting and questioning the Student Model, *Proceedings of World Conference on Artificial Intelligence in Education*, AACE, Washington DC, USA, pp. 501-508.

Brusilovsky, P., Schwarz, O., Weber, G. (1996) ELM-ART: an intelligent tutoring system on the World Wide Web. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.): *Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS-96*. Springer, Berlin, p 261-269.

Calvi L., & Cristea, A. (2002) Towards Generic Adaptive Systems: Analysis of a Case Study. In P. De Bra, P. Brusilovsky, R. Conejo (Eds.), *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga (Spain), May 2002, Springer:LNCS 2347, 79-89.

Cawsey, A J, R B Jones, J Pearson, (2000) The Evaluation of a Personalised Health Information System for Patients with Cancer, *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 10(1): 47-72.

Corbett, A T, and J R Anderson, (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253--278.

Cristea A., & Aroyo, L. (2002) Adaptive Authoring of Adaptive Educational Hypermedia. In P. De Bra, P. Brusilovsky, R. Conejo (Eds.), *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga (Spain), May 2002, Springer:LNCS 2347, 122-132.

Czarkowski, M and J Kay, (2003) How to give the user a sense of control over the personalization of adaptive hypertext? *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, User Modeling 2003 Session*, 121-132. http://wwwis.win.tue.nl/ah2003/proceedings/

Czarkowski, M and J Kay, (2002) A scrutable adaptive hypertext, De Bra, P, P Brusilovsky, R Conejo (eds), *Proceedings of AH'2002, Adaptive Hypertext 2002*, Springer, 384 -- 387.

Czarkowski, M and J Kay, (2001) Tutor: support for scrutably personalised documents, Proceedings of ADCS'2001, *Australian Document Computing Symposium*, 29-36. http://www.ted.cmis.csiro.au/adcs01/

Czarkowski, M and J Kay, (2000) Bringing scrutability to adaptive hypertext teaching, *ITS'2000, Intelligent Tutoring Systems*, Gauthier, G, C Frasson, K VanLehn (eds), Intelligent Tutoring Systems, Springer, 423--432.

De Bra, P, J P Ruiter, (2001) AHA! Adaptive Hypermedia for All, *Proceedings of the WebNet Conference*, 262-268, October 2001.

De Bra, P., Aqerts, A., Berden, B., de Lange, B., Rousseau B., Santic, T., Smits, D., & Stash, N. (2003) AHA! The Adaptive Hypermedia Architecture. *Proc. Int. Conf. on Hypertext and Hypermedia 20*03, Nottingham (UK), 81-84.

Höök, K, J Karlgren, A Waern, N Dahlbäck, C G Jansson, K Karlgren, and B Lemaire, (1996) A Glass Box Approach to Adaptive Hypermedia, *Journal of User Modeling and User Adapted Interaction*, special issue on Adaptive Hypermedia, 6(2-3), 157-184.

Jameson, A., and E Schwarzkopf, Pros and Cons of Controllability: An Empirical Study, De Bra, P, P Brusilovsky, R Conejo (eds), *Proceedings of AH'2002, Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, 193-202.

Kay, J, (2001) Learner control, *User Modeling and User-Adapted Interaction*, Tenth Anniversary Special Issue, 11(1-2), Kluwer, 111-127. http://umuai.informatik.uni-essen.de/anniversary.html

Kobsa, A. (2002) Personalized hypermedia and international privacy, Communications of the ACM archive, Volume 45 , Issue 5 (May 2002), SPECIAL ISSUE: The adaptive web, pp 64 – 67.

Maes, P., and Schneiderman, B. (1997). Direct Manipulation vs. Interface Agents: A Debate. Interactions, Vol. IV Number 6, November 1997, ACM Press, pp 42-61.

Morales, R., Pain, H., Bull, S. and Kay, J. (Eds.). (1999). *Open, interactive, and other overt approaches to learner modelling (Proceedings of the workshop at AIED 99).* International Journal of Artificial Intelligence in Education, 10, 1070-1079.

Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test, *International Journal of Human-Computer Studies*, 41(1-6), 385-397.

Rich, E, (1979) User modeling via stereotypes, *Cognitive Science*, 3, 355-66.

Weber, G. and Brusilovsky, P. (2001) ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education* 12 (4), Special Issue on Special Issue on Adaptive and Intelligent Web-based Educational Systems.