Visualising Narrative Structures in Personalised e-Learning Systems

By

Fionán Peter Williams

A dissertation submitted to the University of Dublin,

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

May 2006

Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: ______________________

Fionán Peter Williams

31st May 2006

Permission to lend and/or copy

I agree that Trinity College Library may lend or copy this dissertation upon request.

Signed: ______________________

Fionán Peter Willaims

31st May 2006

Acknowledgements

I would like to thank my project supervisor, Owen Conlan, for his inspiring direction and advice during this project. On a personal note I would like to dedicate this dissertation to the memory of the late Julian Vereker OBE, founder of NAIM Audio, Ltd., Salisbury, England, who in a memorable 'Loreal Moment' declared that "you are worth it!" He is greatly missed.

Thanks to both my parents, Kieran and Dina, who instilled in me a curious mind.

And finally, to my wife Mary, who has endured the making of this dissertation with love and humour, it's finished!

Abstract

The nature of the files generated by server access logging, and adaptive e-Learning systems, does not lend itself to easy scrutiny or interpretation by humans. The problem thus is to find techniques to represent this data in a way that allows users to quickly assimilate key relationships and acquire knowledge efficiently from it. Information Visualization has emerged as a new field of research with the potential to solve many of the problems associated with representing large data collections.

This dissertation proposes a novel approach to the visualisation of complex, but interrelated, sets of information to ease user cognition. Principally, it provides end users, such as learners, with a means of visualising the complex space in which they learn, with the aim of supporting the development of meta cognitive insight pertaining to their individual learning style. It also provides course authors with a means of understanding how learners used their courses and gives them a means to detect usage patterns and diagnose problems.

A review of the state of the art relating to the visualisation of temporally correlated data set is presented, together with an investigation of the methodology by narrative structures, and learner models, are defined in personalised e-Learning systems. The software implementation and visual design of a prototype visualisation which displays the correlation between learner activity and the adaptive e-Learning course narrative is described. The prototype narrative analyser is evaluated with the aid of a group of expert users, conclusions are drawn, and suggestions for further development and research in this area are presented.

Table of Contents

Table of Figures

Illustration Index

1. Introduction

1.1 Motivation

The nature of the files generated by server access logging, and adaptive e-Learning systems, does not lend itself to easy scrutiny or interpretation by humans. The problem thus is to find techniques to represent this data in a way that allows users to quickly assimilate key relationships and acquire knowledge efficiently from it. Information Visualization has emerged as a new field of research with the potential to solve many of the problems associated with representing large data collections. "For information to become knowledge, we need to interpret and understand it. Visualization in general responds directly to this need" [Carpendale et al, 1997].

Cognitive overload can lead to a "lost in hyperspace" situation where users find themselves disoriented in a richly interrelated information space [Conklin, 1987]. In personalised e-Learning applications the narrative describing the sequence and formulation of the learning experience is personalised on a per user basis. This can lead to those responsible for the administration and authoring of such experiences being inundated with a large volume of uncorrelated data about the usage of their systems, which usually means it is never accessed and is therefore rendered useless.

Through the use of visualisation techniques users of adaptive e-Learning systems can be empowered by providing an 'at a glance' overview of the way they have interacted with the suggested course narrative. This facilitates the development of meta-cognition with regard to their learning style. Course designers can also benefit from developing insight into the performance of the adaptive e-learning system through the correlation of many user models and access log histories with course narratives generated for individual users of the system. However, text based file formats such as XML files, while 'human readable' are not conducive to the rapid assimilation of large sets of information, such as server access logs where the records of the access histories of many users are interleaved in large files. Correlation of these access histories with the suggested narrative structures and subsequent interpretation is a task which has many potential benefits for users, course designers, and those responsible for monitoring the effectiveness of on-line courses.

This dissertation proposes a novel approach to the visualisation of complex, but interrelated, sets of information to ease user cognition. Principally, it provides end users, such as learners, with a means of visualising the complex space in which they learn. It also provides course authors with a means of understanding how learners used their courses and gives them a means to detect usage patterns and diagnose problems. Course authors may find that visualisation techniques assist in the validation of the narrative structures that are matched with individual learners based on an assessment of their preferred learning style, prior knowledge, and learning goals.

1.2 The Research Question

This dissertation explores the relationship between adaptive information and its potential visualizations. This thesis asks whether visualisation techniques as applied to the task of visualising correlated course and access history information can provide personalized e-Learning course authors and administrators with an effective analysis and diagnosis tool. In parallel the potential for utilizing such visualization techniques targeted at actual learners will be investigated. Principally, the use of visualization to support meta-cognition will be discussed.

1.3 Objectives and Goals

The primary goal of this project is to investigate the potential applications of the visualisation of course narrative structures as generated by personalised e-Learning systems correlated with the actual usage histories of individual learners. The visualisation design should be clear and intuitive in order to facilitate rapid assimilation of key relationships and the detection of problems. Access to the visualisation should be simple, without the requirement for application installation and configuration, in order to encourage frequent use. The design should allow for straightforward integration with existing infrastructure, with due consideration given to issues of privacy and security.

The visualisation should be capable of aiding designers and administrators in analysing the effectiveness of the adaptivity implemented in adaptive e-Learning systems. This adaptivity is expressed in individually tailored course narratives generated in response to questionnaires designed to capture an individual's learning style as characterised by the Honey & Mumford

Learning Style Model [Sarrikoski, 2000]. The visualisation of usage patterns by correlation of the system generated course narrative and the actual usage of the course by learners as documented in access log files should allow for validation of the adaptivity implemented in the system and aid the detection of mismatched learner models and course narratives.

To realise this goal the following four objectives were defined:

1. To research and document:

   - The process by which learning style is classified according to the Honey & Mumford system of classification.
   - The methodology supporting the definition of course narrative structures in personalised e-Learning systems.
   - The state of the art in approaches taken to the development of temporally correlated visualisations.

2. To design and implement a prototype narrative analyser.

3. To evaluate the prototype narrative analyser by means of hands on testing and user feedback.

1.4 Dissertation Overview

This introduction chapter is intended to provide an overview of the motivation behind this research project, define the research question to be addressed, and detail the goals and objectives to be pursued.

The next chapter is a review of the state of the art, split into two main sections. The first is an overview of personalised e-Learning systems that explores the relationship between learner models and adaptive behaviour in personalised e-Learning systems. Next is an investigation of the approaches taken to the visualisation temporally correlated data sets, including a description of the PlaceTime concept. The PlaceTime project explored the potential for the development of a library of reusable visualisation components focussed on the correlation of temporal and location data [Hampson, Williams, 2004]. Although PlaceTime revolves around the twin axes of time and location the issue of visualising the correlation between different data sets is central to the issues being researched in this dissertation. This chapter concludes with an overview of the technologies through which the required visualisation can be realised.

Chapter three, *Software implementation*, reviews the implementation of the software components of the prototype narrative analyser using HTML, Javascript, and Scalable Vector Graphics generated at run time. Chapter four, *Visualisation design*, details the visual design of the prototype narrative analyser. In both of these chapters the sources of the data sets to be visualised are considered and design features relating to the prototype narrative analyser presented here are examined.

Chapter five, *Trial and evaluation*, deals with the trial and evaluation of the prototype narrative analyser based on hands on use and a set of exploratory questions posed in individual interviews with the evaluators. Finally, this dissertation is completed by chapter six, *Conclusion*, which presents a review of the objectives defined in chapter one, *Introduction*, and outlines recommendations for future research in this area.

2. State of the Art

2.1 Introduction

This chapter logically falls into two parts. The first part, beginning with section 2.2, *Narrative and the learner in personalised e-Learning systems*, explores the definition of learning style according to the Honey & Mumford learning styles classification, and will detail the process through which course narratives are constructed in personalised e-Learning systems as they reconcile learning style models, prior knowledge and learning goals with the available pool of learning objects. A particular focus is placed on the APeLS (Adaptive Personalised e-Learning Service) system developed by the Knowledge and Data Engineering Group in the Department of Computer Science at Trinity College, Dublin. Also considered is the approach to narrative construction taken by the AHA! (Adaptive Hypermedia Architecture) system developed in the Department of Computer Science at the Eindhoven University of Technology, Holland and the 3DE (Design Development and Delivery Electronic Environment Educational Multimedia) system developed as part of a research project which formed part of the European Union IST 5th Framework Programme.

The second logical part of this chapter begins with section 2.3, *Temporal Visualisation Techniques*, and explores the approaches taken to the visualisation of temporally correlated information, supplemented in section 2.4, *PlaceTime Visualisation*, by an overview of the visualisation techniques proposed in the PlaceTime project [Hampson, Williams, 2004]. Section 2.5, *Implementation Technologies*, summarises the available implementation technologies suitable for the implementation of a narrative analyser prototype. The chapter concludes with a summary of the investigation of the state of the art.

2.2 Narrative and the learner in personalised e-Learning systems

This dissertation has set out to investigate whether developing meta cognition with regard to their own approach to learning can enable learners to redefine their learning style more accurately, thus improving the effectiveness of the adaptivity applied by the personalised e-Learning system. User updating of learner models and the subsequent redefinition of the course narrative with reference to the updated learner model is encouraged in some personalised e-Learning systems, such as the APeLS system examined later, while other systems apply adaptivity through system updates to the learner model based on monitoring learner accesses to each learning object and evaluating the match between the learner's current learning style and the learning style meta data encapsulated in the learning object.

2.2.1 Modelling the learner – the Honey & Mumford learning style classification.

The most important element of an e-Learning system is how precisely the system models the learner [Conlan et al, 2003]. Before the currently dominant constructivist learning paradigm became established the two dominant paradigms were the behavioural and cognitive views of learning [Sarrikoski et al, 2000]. The behavioural model proposed that a change in observable behaviour was the outcome of learning and modelled relations between attributes of the learner including intelligence, abilities, and social background, validated by quantitative measurements such as exam grades. The cognitive model began to replace the behavioural model after becoming popular in the 1960s as researchers became more interested in the process of learning itself, examining topics such as reasoning, comprehension, and problem solving [Canavan, 2004]. The constructivist view is concerned with "the learner's own active contribution to his learning process in a social context where the learner constructs his knowledge by combining new information and experiences with his existing knowledge and structures"[Sarrikosky et al, 2000].

Two distinctive styles of learning, one based on formal structured activities such as reading a book or attending lectures, and the other based on learning through experience, were identified with the first type being more familiar and more straightforward than experimental learning [Honey et al, 1992]. However, although there is extensive research on learning styles there is no agreement or acceptance of any one theory [Bruen et al, 2002].

Four distinct stages of the learning process were presented by Honey and Mumford [Honey et al, 1992], building on the work of David Kolb. These are:

1. Having an experience.
2. Reviewing the experience.

3.Concluding from the experience.

4.Planning the next steps.

It is not required that a learner starts at step one and works through the steps to step four. Starting at step two with some newly discovered knowledge one could arrive at conclusion and define how to apply this knowledge. Four learning styles are associated with the four stages of learning identified above. They are:

1.Having an experience. Learning style – Activist. Activists are open minded, not sceptical, and prepared to act first and consider the consequences later. They are happy to work in a team but want to be the centre of attention. They learn best when solving new problems, are involved in new experiences, or presented with new opportunities.

2.Reviewing the experience. Learning style – Reflector. Reflectors are comfortable when able to stand back and observe. Naturally cautious, they keep a low profile and may be perceived as slightly aloof, or distant, but unflappable. They like to observe, and then have time to reflect on what happened. They tend not to like tight deadlines.

3.Concluding from the experience. Learning style – Theorist. Theorists create logically sound theories by adapting and integrating observations. Like reflectors they can be detached but are analytical in a logical manner. They like to be in situations where they can apply their knowledge with a clear purpose, or to be able to assimilate new ideas or concepts even if they are not applicable immediately. They like to understand the ideas behind things.

4.Planning the next steps. Learning style – Pragmatist. Pragmatists like to test new theories in practice. They like to get things done and act confidently on ideas that interest them. They like solving problems in a practical way. They learn best when the topic and the job at hand are closely linked and like to get feedback in situations such as role playing. They are attracted by new ideas or techniques that have clear advantages.

Most people have developed a preference for one of the stages of learning that may lead to one stage being given greater emphasis with a detrimental effect on the others [Honey et al, 1992]. An individual's learning style is categorised by answering a set of questions presented on the Honey & Mumford learning style questionnaire. The full questionnaire consists of eighty questions split into 20 for each learning style and was developed to capture the learner's learning style through indirect questions such as “I actively seek out new experiences” or “In discussions I like to get to the point". The approach taken is to probe behavioural tendencies and not learning tendencies because it is believed that people do not consciously consider how they learn and therefore it is not helpful to ask questions that directly enquire into this [Honey et al, 1992].

A shortened version of the questionnaire has been implemented on the APeLS system. Each question has four possible answers ranging from 1 -"I disagree", to 4-"I fully agree". The learner profile is then expressed by a normalised parameter for each of the four learning styles which ranges from 0 – 100, with 0 being a low preference for the style and 100 representing a high preference. The learners preferred learning style is then determined from the highest of the values recorded [Canavan, 2004].

2.2.2 Developing a narrative for the learner.

Arriving at a personalised narrative in the APeLS architecture is based on a number of steps. The domain expert creates a CCG narrative (Content Candidate Group), which is abstracted from the individual learning objects. "The main goal of the multi-model approach used in APeLS is to separate the learning content from the adaptive linking knowledge or narrative"[Conlan et al 2002]. The separation of the learning content from the narrative supports the reuse of content by potentially allowing the learning object to be included in many possible narratives.

Course authors can create different narratives using the ACCT interface (Adaptive Course Construction Toolkit) which achieve the same learning objectives but are based on alternative pedagogical models e.g. a didactic structure or a case based approach. Through the sequencing of the narrative, courses can be generated that differ in ethos, learning goals, pedagogical approach, and learner prior experience [Conlan et al, 2003]. The narrative's primary goals are to produce courses that are structured coherently and satisfy the learner's goals in a way that engages the learner [Dagger et al, 2003].

The most appropriate narrative is selected at runtime by the adaptive engine calling a candidate selector which chooses the most appropriate narrative from the available candidate narrative group based on the learner model [Conlan et al, 2002]. Although the APeLS system can support any learning style model in theory, the current model in use is the Honey & Mumford one.

Adding support for another model requires adding appropriate meta data to the learning objects and implementing a new candidate selector for that type.

Once the course narrative has been defined at runtime at the start of a session in APeLS it tends to be quite stable [Canavan, 2004]. This perceived stability might reflect the learner's lack of meta cognition regarding their learning style as it is possible to update their learning style and thus redefine the narrative. On the other hand it may reflect the effectiveness of the adaptivity applied, with no further changes required.

AHA! Avoids the definition of learning style by questionnaire as (the designers argue) it can lead to learners being assigned to stereotypical groups and assumptions about their learning style are not updated during the learner's interactions with the system. AHA! Aims to infer the learner's learning style through the monitoring of their browsing behaviour [Stash et al, 2004]. Learners may select a learning style from a drop down list and if they then access the recommended concept then the system's confidence that the learning style was correctly identified is increased. Conversely, if an inappropriate or undesirable concept is accessed then the confidence level is reduced. If the confidence level falls below a threshold level then the learner is asked if they would like to choose another learning style. If no learning style was selected then the system can match the learner with a style based on the same mechanism.

Because of this 'object by object' refinement of the learning style model the narrative structure is to some extent fluid. To avoid the learner following a fragmented narrative whereby the same piece of content is presented differently each time it is accessed due to the adaptive presentation mechanisms employed, AHA! incorporates a configuration option which defines the 'stability' of the presentation. A page can be configured as always adapted or always stable. Unlike APeLS which has a clearly defined separation of the Learner, Narrative, and Content models in order to facilitate the reuse of content, AHA! intertwines the domain and adaptation models.

The 3DE system generates a learner model based on the shortened form of the Honey & Mumford questionnaire. Only the Honey & Mumford learning style model is supported in 3DE but this restriction does have the advantage that all learning objects share the same references to learning style, thus allowing them to be reused without modification. The learner model is used to match the courses available with the learner's individual learning style. Before the course narrative is constructed, the learner may choose to use a version of the course that matches their learning style, or choose a different learning style. Once a course narrative is defined at the start of a session it will not change.

In 3DE content is arranged in a structured hierarchy in order to facilitate reuse. Also available in the 3DE environment is a 'learning to learn unit'. The developers recognise that once a learner is aware of their learning style that they can use this knowledge to improve their learning skills. This unit can be accessed from any course and the student can get information about learning skills and techniques which may be suited to their learning style [Del Corso et al, 2003].

2.3 Temporal Visualisation Techniques

Information Visualisation encompasses a wide area of research that is being seen by many as the answer to displaying large amounts of information in a useful and accessible way. The growth of the internet, the computerisation of business and defence and the deployment of data warehouses have created a widespread need, and growing appreciation for information visualization [Breiteneder et al, 2002]. Temporal visualisations of data have been among the more popular techniques used by researchers in this field, and in this section we will take a look at some examples and their relevance to narrative analysis.

"Graphical displays of data as it occurs over time is one of the most common and powerful methods of visualising information and have been in continuous use for the past 200 years" [Tufte, 1983]. Most multimedia and audiovisual applications such as Macromedia Director and Adobe Premier use the time line metaphor, as it is a familiar and intuitive way to interpret, edit and synchronise temporal elements. Apart from countless commercial applications, a number of research projects have also used the time line as a key component of their visualisation techniques. One such system is LifeLines, a research project developed at the University of Maryland. In essence, LifeLines is "a general visualization environment for personal histories that can be applied to medical and court records, professional histories and other types of biographical data" [Heller et al, 1998]. For instance, when LifeLines is used to display a medical record of a patient, their entire medical history is displayed on the time line with the user able to change the scale of the view in order to focus on particular details. Icons, horizontal lines, colour, and line thickness indicate events and relationships.

Results from experiments on LifeLines suggest that overall, “users are better able to comprehend and remember the information presented by the LifeLines visualization than with a tabular representation” [Geisler, 1998]. Like PlaceTime, which is investigated in the next section, it utilises time lines, icons and colour to encode information, and the form of encoding employed means that many different domains can be represented by it. The designer's aim of creating a set of generic tools that can be utilised by a range of applications is comparable to the goal set for PlaceTime.

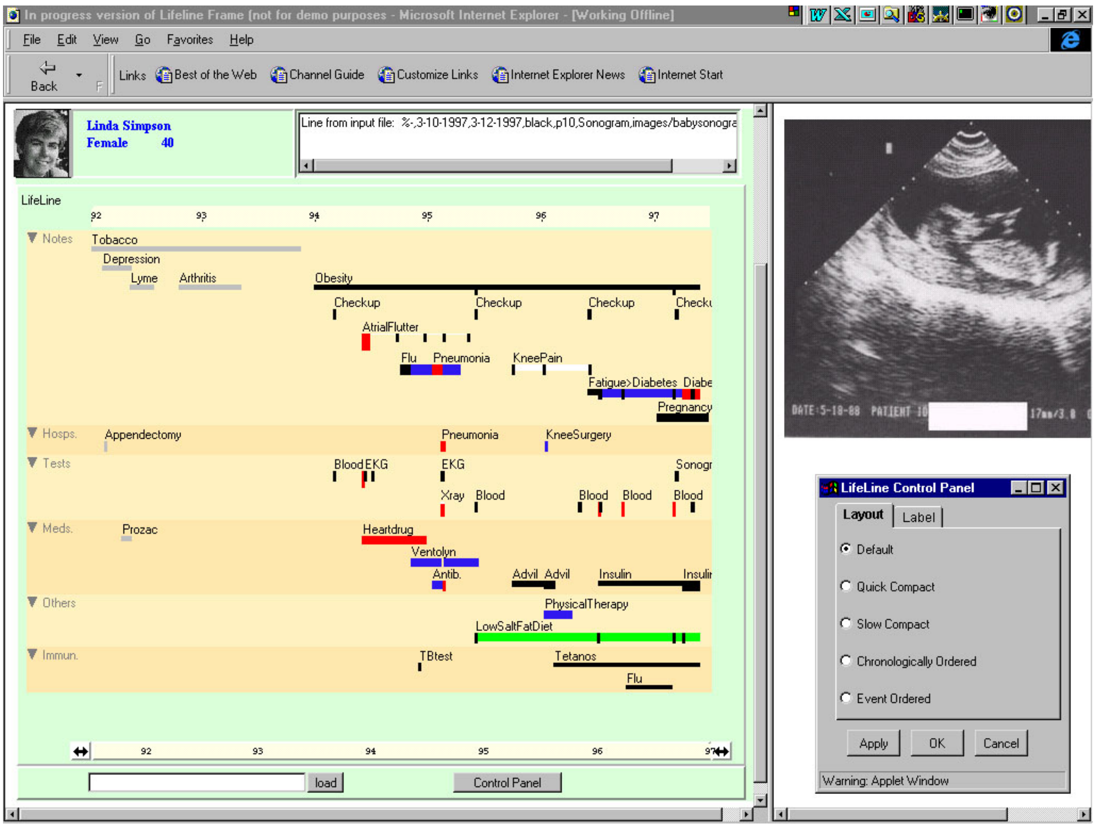


Illustration 1: LifeLines Screenshot

Another research project that used a time line metaphor to represent information is ThemeRiver [Havre et al. 2000]. Its major design goal “was to provide a visualization of theme change over time” using the metaphor of a river to achieve this. Fundamentally, a collection of documents (for instance newspaper articles over a certain time period) are examined by ThemeRiver, with key changes in themes deciphered by observing changes in the ThemeRiver over time. The ThemeRiver consists of thematic streams (representing a key term and differentiated by colour) and external news events that are displayed along the time line.

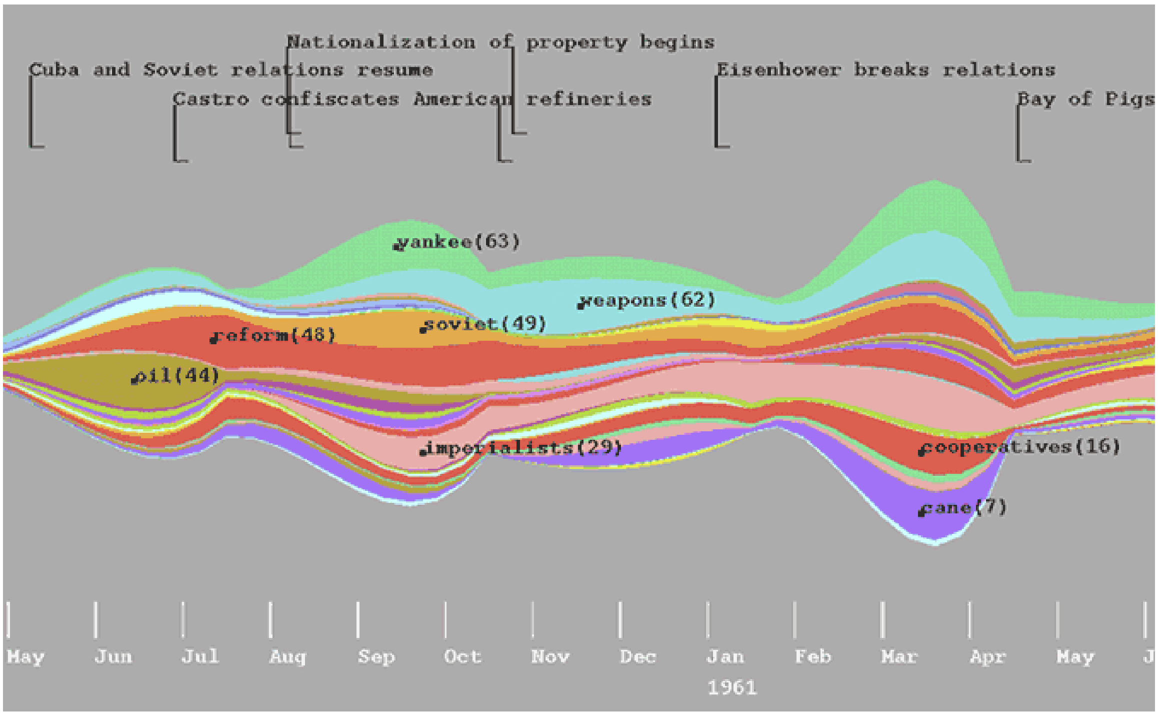


Illustration 2: ThemeRiver Screenshot

Any change in the width of a thematic stream corresponds to the frequency the term occurs in the news. Thus in this instance it is possible to correlate external events with a thematic shift in the news. It is their assertion that any abrupt changes in theme are much easier to locate in ThemeRiver than in an equivalent text based system, however there are some major limitations in its design. For instance, as with a lot of other visualisation systems ThemeRiver lacks a dedicated component that "can either filter out noises or amplify signals in the original data". Chen believes this is partly due to "an overly emphasised reliance on the perceptual and cognitive abilities of human beings" [Chen, 2004].

The final project we will mention that uses the time line metaphor is TimeSpace [Jones & Krishnan, 2005], which describes itself as an activity based temporal visualisation of personal information spaces. TimeSpace can be used alongside or in place of current systems (Microsoft

Windows for instance) to display users personal files in a non-hierarchical manner. Within TimeSpace there are two main interactive visualisations, one that shows an overview of the users' activities along a time line, and one that presents a detailed view of the files in each activity and their development.

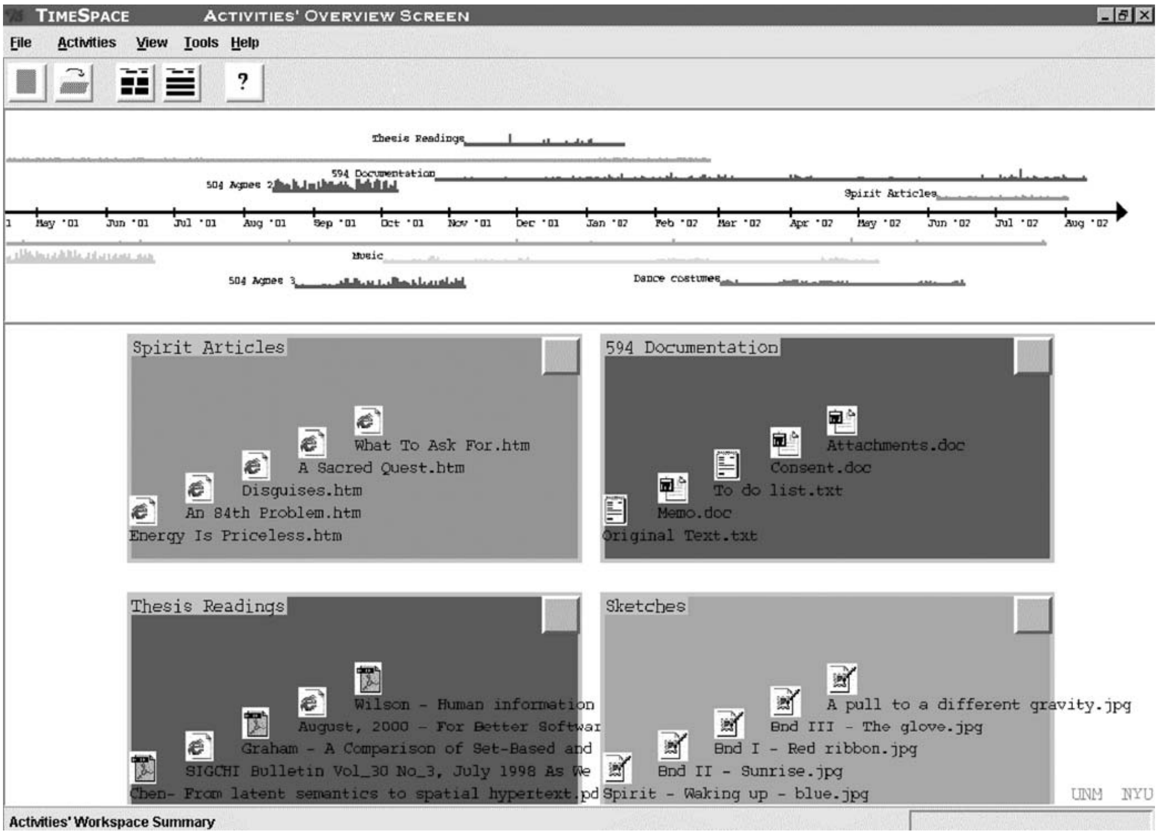


Illustration 3: TimeSpace Screenshot

Users can pan and zoom to focus in on particular details of interest, and direct manipulation is permitted, which eases some navigation problems associated with large document sets. Observational studies on the use of the system revealed positive views on the temporal metaphor with many finding the visualisations provided "a context for their work… and an overview of all their work in progress". TimeSpace is another example of how powerful the time line metaphor can be as a visualisation tool. Furthermore, studies of its use highlight the potential for users to gain insights into large data sets when they are presented to them in an alternative way to that which they have become used to.

"The core of information visualization is finding a way of visually representing information in a manner that is most effective and pleasing for user comprehension. This involves mapping data values onto visual parameters. Our goal is to automatically provide the best mapping given a certain data set and a number of different visual metaphors" [Abel et al, 2000]. This statement defines the motivation behind the creation of new visualisation techniques and informs the approach taken to the design of the prototype narrative analyser presented in this dissertation.

2.4 PlaceTime visualisation

2.4.1 Overview

This section explores the main features of the PlaceTime visualisation design, which merges both temporal and location based information. The techniques developed for PlaceTime have direct relevance to the design of the prototype narrative analyser.

Currently, most visualisations are 'temporally static' - they offer a view of the data as it is now, but offer little insight into either the history of the data, the location of the data, or the probable course of future events related to the data. Therefore the PlaceTime project set out to create a generic set of display components and display rules for the display and editing of the temporal and location relationships of elements in a database.

An implementation of PlaceTime would support tailoring locally, at run time, to a wide variety of interface applications in the Ubicom domain by means of an interface definition encapsulated in XML data. A visualiser component configured for the target environment creates and manages a display space that is populated from the database using display rules associated with data elements, rather than requiring the visualiser to have knowledge of every possible combination of display rules.

The concept extends to the implementation of interfaces of widely differing scale – from something as small as a 'smart room' wall-mounted control panel, through PC based browser/editors, to full immersion or augmented reality virtual displays. This overview focuses on the application of PlaceTime to a PC based browser and editor.

The display metaphor is based on a concept of the visualiser being 'biased' towards either Time or Location. Events in PlaceTime are defined as being discrete datums representing location or temporal information. Thus, two types of non-exclusive displays are available which aim to clarify events based on their current, past, and future location or the temporal relationships between either one set of serial events or parallel sets of serial events. As stated, both types of display are non-exclusive, meaning that a location-based display can also include temporal information, and vice versa.

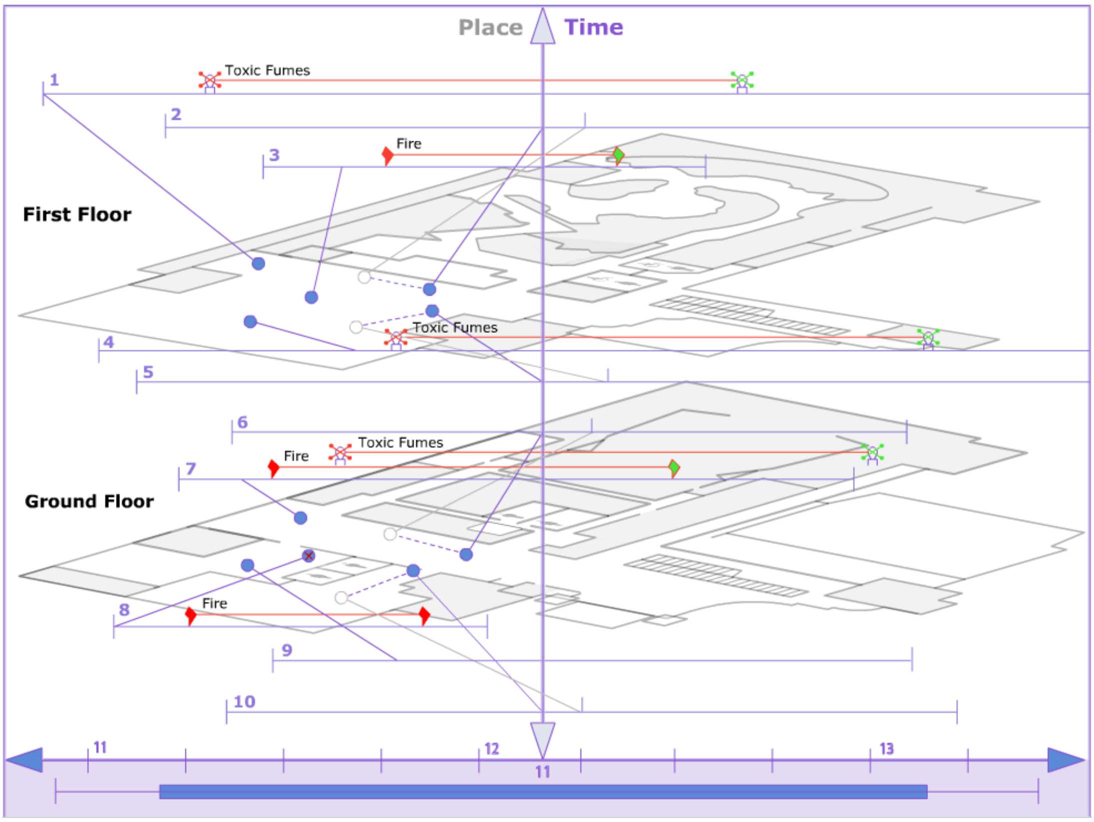


Illustration 4: PlaceTime showing a Time dominant view

Time is defined as flowing from right to left of the display so that scrolling left would be to display more of the past events and scrolling right would display more of the future scheduled or probable events. The interface supports zooming based on Scalable Vector Graphics. It is envisaged that complete display configurations can be saved and restored such that multiple views of the same data can be explored. Configurable view filters include zoom factors, ranges, event types, event grouping, and view angles for both the foreground and background planes.

2.4.2 Interface components and features

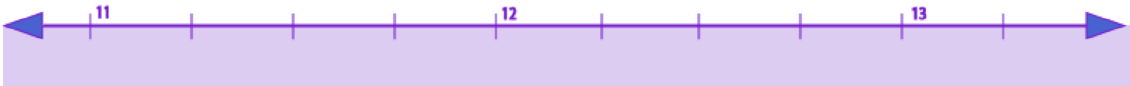


Illustration 5: PlaceTime time bar

The Time Bar reflects the selected time range in two potential states of the viewer. Live mode is when the clock is ticking in real-time and the time scale scrolls right to left at a rate commensurate with the current zoom level. However, if the viewer is off-line the user has control of starting and stopping time. Left-clicking with the mouse on either the left or right hand direction arrows scrolls the view forward or backwards in the available time range. 'Time warping' allows the user to compress the view of either past or future time in order to optimise the displayed region to best exploit the available screen real estate. Warp can be logarithmic or linear according to preferences set.

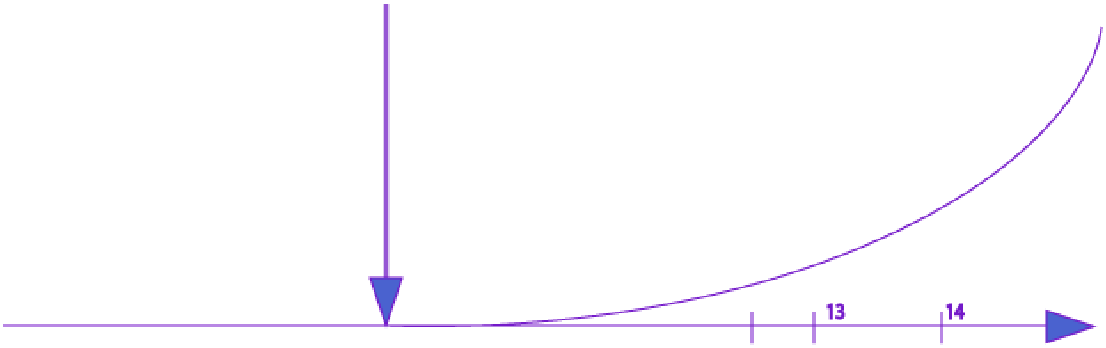


Illustration 6: PlaceTime time warp display

Feedback about the current time warp setting is given by a curved line diverging from the time bar, with an origin at the midpoint of the visible time bar, by an amount relative to the degree of warp.

Illustration 7: PlaceTime view space bar

The View Space bar gives feedback about the current view in relation to the total available time range. Left-clicking the view space indicator and dragging left or right moves the view forward or backwards relative to the available time range.

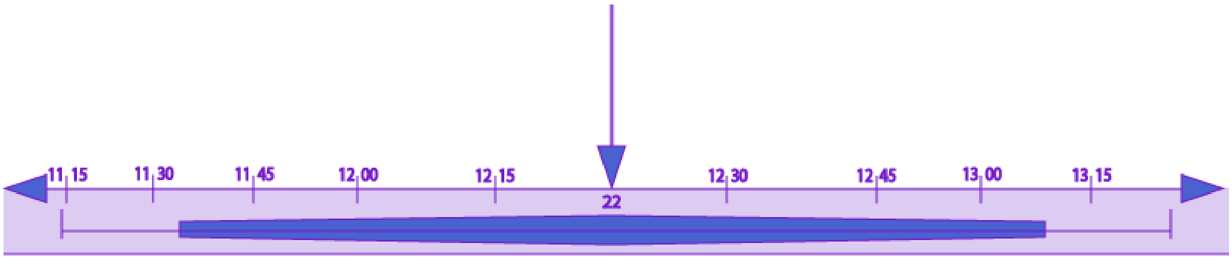


Illustration 8: PlaceTime lensing amount display

Lensing allows the user to expand the view of time around the mid line of the display in order to clarify areas of dense detail. Feedback about the current lensing setting is given by the degree of vertical distortion around the centre line of the view space indicator.

The Now line refers to the current time. Individual time lines are displayed relative to the now line. Movement or actions which take a displayable amount of time to complete are referenced to the Now line. Everything to the left of the Now line is in the past; everything to the right is in the future. In addition the Now line allows for scaling of the view in the vertical axis. The vertical axis refers to the available range of time lines or locations that can be displayed. If the current view settings can not display all available time lines or locations an Event Space View bar is superimposed on the Now line, and the up/down direction arrows are made active. Left-clicking on the up or down arrows or left-clicking and dragging the Event Space bar up or down will slide the visible view through the available range.

Lensing allows the vertical view space to be magnified around the centre of the display. Finally, left-clicking the Time or Place labels switches the display to the Time or Place dominant modes. Left-clicking anywhere on the Now Line switches the display to Time dominant mode.

A 'Local Now Line' can be created by right-clicking on the Now Line and dragging a Local Now Line to the desired time. Right-clicking the Time Bar and dragging over the desired range can select a Time Range. When a Time Range is selected the option of Time Playback is available. When selected this mode causes the Local Now Line to traverse the Time Range selected in real-time or at a rate selected by user preference.

Illustration 9: A single blank PlaceTime time line

Time lines are the heart of the Time Dominant mode. They have a start time but may or may not have an end time. Start and End markers are vertical lines. An identifying label is displayed at the beginning of the time line or the left-most extent of the visible part of the time line if the view is zoomed. A time line may refer to an event or a range of events and reference one or more locations. Time lines that cross the Now line can have both historical and future events.

In Place dominant mode current locations are displayed with a label relative to the location map.

Path information is displayed if an object is in motion. If the object has a destination the projected path is displayed by means of a target location connected by a dashed line. If a destination is specified then an estimate of the arrival time at the destination is displayed as a graph relative to the Now Line. As the object approaches its destination the subtended angle between the two lines connecting the present and target locations to the Now Line becomes more acute until they merge and disappear. If the object is in motion but no destination has been specified then the object displays a dashed line trail.

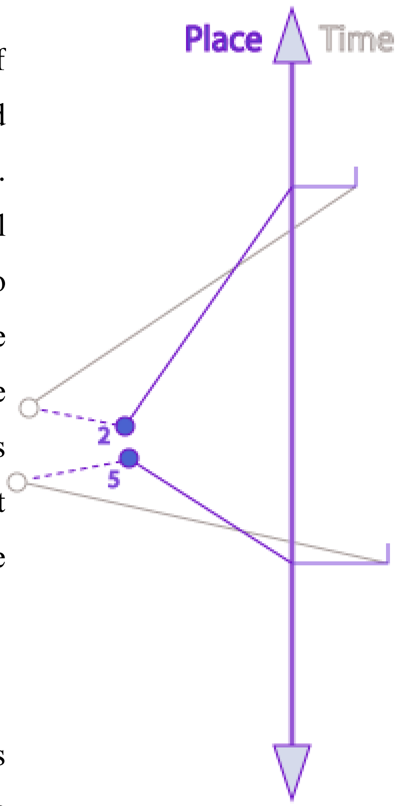


Illustration 10: PlaceTime movement vector display

Mousing over a location causes the path from the object's origin to be displayed. Way points on the path can also be set as a series of targets by repeatedly click-dragging the target locator.

Illustration 11: PlaceTime event display

Events in Time dominant mode can be either anchored at a specific instant in time or have a duration. Events are displayed stacked relative to their parent time line if they have overlapping child time lines, or serially along their parent time line if there is no overlap. Events can have icons associated with them for start and end states. Zooming the display in allows more detail to be resolved for events. Continuously updating readings or streaming media could be displayed as graphs or image sequences superimposed on the time line.

Future events can be either predicted events or scheduled events. Predicted events may be mutually exclusive. For instance “if event 'a' happens then event 'b' will not happen. Branching the future time line shows such exclusive or conditional predictions. The most likely events are on the time line closest to the centre line of the parent time line.

Illustration 12: PlaceTime conditional branching

Events in Place dominant mode are displayed relative to the location map. Events are displayed at their associated location and can have both icons and labels displayed according to user preferences.

Illustration 13: PlaceTime events

Events are also displayed relative to the Now bar – i.e. visible or not depending on their status at the current value of the Now bar or Local Now bar. If a Local Now or Time Range is selected the events active at the Local Now setting or encompassed in the Time Range are displayed. If playback mode is selected for a Time Range the events displayed are relative to the Local Now bar and are thus animated.

Colour is employed to emphasise the separation of the foreground plane from the background. Each layer is given a colour tone that sets it apart from the other visible layers. Alternative colour schemes may be applied but they must conform to the display rules. Time lines can also display status information through colour. A red flashing time line might signify that it is 'ready to record events' and solid or undulating red tones may indicate 'recording of events in progress'.

Colour can also be employed as a cue to the current view location relative to the Now line. Future events might have a 'blue shift' and past events a 'red shift' of a degree dependant on the relationship of the offset from the Now line to the scope of the available time space. Transparency can be used to allow occluded events to be visible when the background planes are covered by elements of the interface.

Display rules identified for a display manager component in an implementation of PlaceTime include:

- Time lines must be separated by enough vertical space for their child event time lines or branching predictions to be clearly defined.
- Time lines should be automatically grouped according to location, common events, or user defined preferences.
- Maximum densities of displayed data are determined according to current zoom factor
- Time lines should subtend a 20˚ angle to their point of location to allow a clear relationship to be expressed.

The display manager would also handle level of detail switching relative to the zoom factor. For instance when enabling the display of continuous sensor data or streaming data when zoomed in.

2.5 Implementation technologies

The software implementation technologies chosen for the development of a prototype narrative analyser should require no user installation as mandated in the objectives defined in chapter one, *Introduction.* Ideally, the chosen technologies should make use of established freely available technologies such as JavaScript or ECMA Script scripting languages together with emerging technologies designed to support the semantic web, including Scalable Vector Graphics (SVG) and the X3D object definition and behaviour language.

The clear alternative to the non-proprietary SVG and X3D technologies is the Flash animation platform developed by MacroMedia. However, Flash does not support extension into the third display dimension, which may be required to display complex correlations, and is of course a proprietary solution requiring the Flash player to be installed in the client browser.

Basing the visualiser on JavaScript and SVG would allow for a flexible environment where scaling factors can be managed 'on the fly', an essential feature as narrative lengths and access volumes can not be determined in advance. Basing the narrative analyser display on open, widely supported human readable data formats makes it possible for other applications to benefit from any data organisation generated through the use of PlaceTime. This is consistent with the vision of the semantic web, whereby each layer of meta data added as new XML tags adds to the richness of knowledge about data elements.

Java would be the language of choice for the implementation due to the availability of libraries and interfaces for SVG, X3D parsers, and support for a wide range of execution environments, the latter important due to the requirement to allow for the implementation of the visualiser component in different configurations. However, this would necessitate a user installation of the client application. It may be that a visualiser component of the course composition tool kit would benefit from a Java implementation due to the file I/O and the extensive interfaces available to databases and other data sources.

Server side scripting or EJB components could be used as part of a wider deployable version of the narrative analyser and could enable useful data compression of the files needed by the client visualiser through XSLT transforms which retain only the data relevant to the currently selected learner from the access log files.

2.6 Summary

This chapter started by investigating how learning styles are defined using the Honey & Mumford learning style classification in section 2.2.1, *Modelling the Learner*. Next, in section 2.2.2, *Developing a narrative for the learner*, the methodology behind the composition of personalised course narratives in personalised e-Learning systems was researched, with a primary focus on the APeLS system, and consideration given to the narrative composition approaches taken by the AHA! and 3DE systems.

Section 2.3, *Temporal Visualisation Techniques*, examines current techniques applied to the visualisation of temporally correlated data sets. Section 2.4, *PlaceTime visualisation*, provides an overview of the visual design of PlaceTime, an interface which merges location and temporal data, and which can inform the visual design of a prototype narrative analyser.

Section 2.5 provides a brief overview of the implementation technologies that could be utilised in the development of a prototype narrative analyser. It was noted in this section that a learner centric visualisation could be implemented using JavaScript and SVG, but that a course composition tool kit version of a narrative analyser would require the functionality and broad range of data and file interfaces supported in a mainstream computer programming language such as Java.

Objective 1 for this dissertation has been achieved, which was defined in chapter one, *Introduction*, as:

1. To research and document:
    - The process by which learning style is classified according to the Honey & Mumford system of classification.
    - The methodology supporting the definition of course narrative structures in personalised

e-Learning systems.

- The state of the art in approaches taken to the development of temporally correlated visualisations.

This allows the next stage of the design and development of a prototype narrative analyser to proceed to the design and implementation stage, pursuant to the objective of determining an answer to the research question defined in chapter one, *Introduction*.

3. Software implementation.

3.1 Introduction

Chapter three will review the software implementation designed to support the prototype narrative analyser, employing software technologies including HTML, JavaScript, and Scalable Vector Graphics, chosen based on the discussion presented in section 2.5, *Implementation Technologies*, of chapter two, *The State of the Art.* A visualisation design and implementation, detailed in chapter four, *Visualisation Design,* will be completed in accordance with the achievement of objective two defined in chapter one, *Introduction*, which is - “To design and implement a prototype narrative analyser”.

The completed prototype narrative analyser will be utilised as the focus of chapter five, *Trial and Evaluation*, in which the research questions driving this dissertation will be addressed. This 'installation free' form of the narrative analyser is ideally suited to the role of realising a visualisation accessible to learners, while a Java based version featuring file I/O functionality may be better suited to the potential role for narrative analysis as a component in the suite of course composition tools.

3.2 Data sources and software components

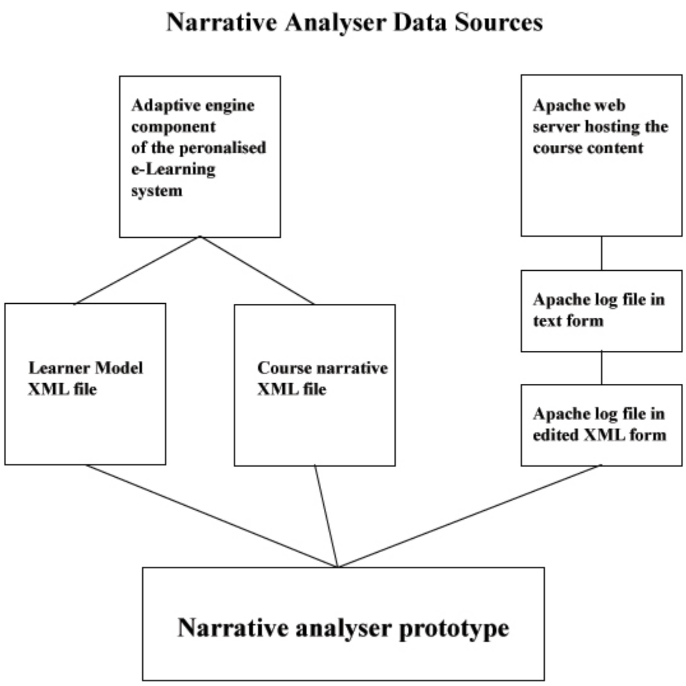


Illustration 14: Narrative analyser data sources

Files from which data pertaining to the visualisation of narrative structures are drawn include the *Learner Model,* and the *Course Narrative* XML files generated at run time by the adaptive engine component of the APeLS personalised e-Learning system. A second third data source is the log files generated by the Apache web server that delivers course content to learners. It is from these log files that the correlation with the suggested course narrative can be determined.

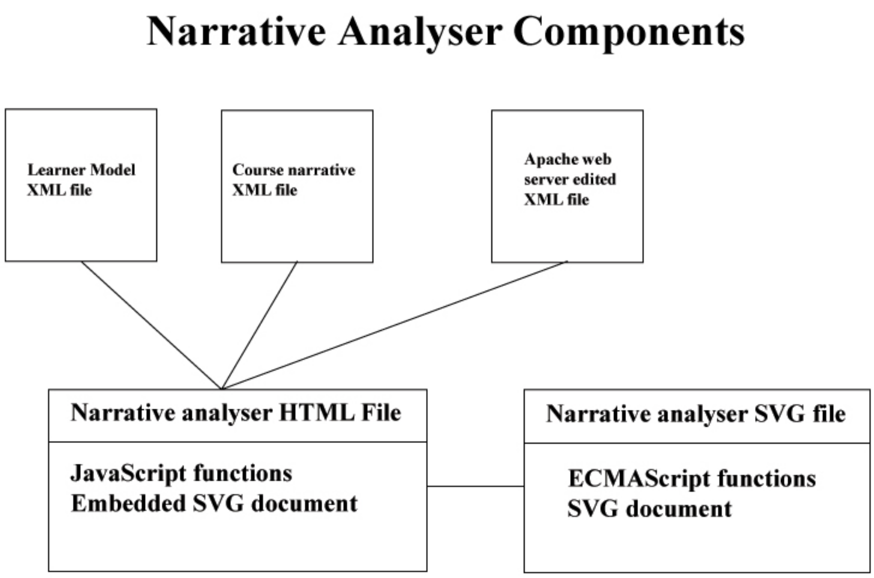


Illustration 15: Narrative analyser components

The prototype narrative analyser software components comprise of a HTML file which defines the structure of the narrative analyser display and JavaScript functionality which manages the data sources, maintains multiple DOM trees for the selected data source XML files, and implements most of the logic. A Scalable Vector Graphics document embedded in the HTML file allows a separate SVG definition to be activated. The narrative analyser SVG file contains the default definition of the background visual elements, together with ECMA Script functionality and structural elements used to support the dynamic updating of the SVG DOM at run time.

3.3 Software design assumptions

Key assumptions on which the prototype is based include:

1. Subsections are uniquely identified within their section. Duplicate names within the scope of one subsection are not catered for in this implementation.
2. All accesses represented for the four simulated learners happen over a one-month period. The implementation of calendaring functionality in the time line display area was deemed to be outside the scope of this project.

3.4 File structure and process connections

The first step in the implementation of the prototype narrative analyser was to locate and examine the data sources available. Course narrative definitions and learner type definitions are available in discrete XML files as outputs from the adaptive engine component of the personalised e-Learning system.

The access log files are, however, in Apache log text format. They were translated to XML by processing them with a utility called Exchange XML Editor, which resulted in an XML file where the <request> tag contained most of the information required to trace narrative events. Parsing the <request> tags and extending the access log XML file through the addition of extra tags and values accessible from the DOM can be achieved by a number of methods including XSLT and PHP scripts. This was felt to be outside the scope of the requirements of the prototype narrative analyser and it was decided to create a log file by hand which could be tailored to highlight some of the interesting possibilities for narrative analysis.

```
<visit date="25/Nov/2004:13:10:49 +0000">
    <cmd>GET</cmd>
    <request>/sql2/page.jsp?learner=declan&amp;course=SQL
        %20Course&amp;section=Database%20Concepts
        &amp;subsection=Introduction&amp;pagelet=2</request>
    <httpver>HTTP/1.0</httpver>
    <httpcode>0</httpcode>
    <referrer>G</referrer>
    <agent/>
</visit>
```

Illustration 16: A single access event from the log file before editing

First four learners, 'Peter', 'Owen', 'Dave', and 'Declan', were assigned characteristics of distinct learning styles according to the Honey & Mumford learning style classification and their learner model files edited appropriately. Then many interleaved access log entries were created for each learner to form the basis of an analysis in the prototype narrative analyser.

```
<visit date="02/Dec/2004:09:33:35 +0000">
    <cmd>GET</cmd>
    <request>/sql2/page.jsp?learner=peter&amp;course=SQL%20
        Course&amp;section=Database%20Concepts&amp;
        subsection=The%20Relational%20Model&amp;pagelet=3</request>
    <peterAccess>
        <name>Peter</name>
        <section>Database Concepts</section>
        <subsection>The Relational Model</subsection>
        <pagelet>3</pagelet>
        <day>02</day>
        <month>12</month>
        <year>04</year>
        <hour>09</hour>
        <minute>33</minute>
    </peterAccess>
    <httpver>HTTP/1.0</httpver>
    <httpcode>0</httpcode>
    <referrer>G</referrer>
    <agent/>
</visit>
```

Illustration 17: A single access event in the edited access log XML file

An XSLT transform could be applied now to combine all the relevant details for each learner into one file. This transformed digest file would facilitate remote web accesses by compressing the amount of data transmitted to the client browser and would also have the added security benefit of limiting the amount of raw data exposed to client browsers.

The prototype narrative analyser application itself begins when the HTML page containing the main JavaScript application is loaded into a compatible browser. The <BODY> section of the HTML defines the basic layout of the narrative analyser starting with the definition of the three pull down menus – 'Select Learner', 'Select Narrative Display Type', and 'Select Name Display Type'.

The SVG display itself is configured using the <EMBED> HTML tag where parameters of the SVG window are set including WIDTH, HEIGHT, and alignment are set. In the prototype narrative analyser there are no more HTML elements and the bulk of the page is given over to the dynamic SVG window.

```
<body>
    <TABLE width="1024"  BORDER="1" align="center"
        CELLPADDING="0" CELLSPACING="0" bordercolor="#0000
        <TR>
            <td bgcolor="#999999">
                <form name="settings" ACTION="" >
                <SELECT NAME="selectLearner" onChange="setSetti
                    <option selected> Select Learner
                    <option value="Peter">Peter
                    <option value="Owen">Owen
                    <option value="Dave">Dave
                    <option value="Declan">Declan
                </SELECT>
                <SELECT NAME="selectDisplayType" onChange="set
                    <option selected> Select Narrative Display Type
                    <option value="1">Course narrative display
                    <option value="2">Access log narrative display
                </SELECT>
                <SELECT NAME="selectNameDisplay" onChange="s
                    <option selected> Select Name Display Type
                    <option value="1">Show all names
                    <option value="2">Show jump and break names
                    <option value="3">Hide names
                </SELECT>
                </form>
            </td>
        </TR>
        <TR>
            <td bgcolor="#000000">
                <EMBED WIDTH="1024" HEIGHT="600" Align="lef
                SRC="FPW SVG for test 1.svg" NAME="animated_m
            </td>
        </TR>
    </table>
</body>
</html>
```

Illustration 18: HTML <body> section with SVG display

When selections are made in any of the three pull down menus the setSettings() function is invoked which results in the SVG display being updated according to the settings of all three menus. This is the main entry point to the narrative analyser.

Next, inside the narrative analyser application itself the individual files are opened using the Microsoft ActiveX DOM object. If the browser does not support ActiveX objects then the user is shown an error message and the application will not execute further. The application now has a DOM model for each of the XML files required and analysis of their content can be carried out in order to construct the Honey & Mumford display, the narrative display, and the time line display.

```
function loadXMLfile(fileToLoad)
{
	if (window.ActiveXObject)
	{
		loadedFile = new ActiveXObject("Microsoft.XMLDOM");
		loadedFile.async=false;
		loadedFile.load(fileToLoad);
		return (loadedFile);
	}
	else
	{
		alert('Your browser cannot handle this script');
	}
}
```

Illustration 19: The loadXMLfile function

As the prototype narrative analyser is constructed from a number of separate files containing executable scripts, it is necessary for functions to be callable externally. The Adobe SVG plug-in supports external access to functions in the SVG document from the HTML document by declaring the function mappings in the script section of the SVG document. The capitalisation of the first letter of the function name is an indicator that a function being called is external to the current JavaScript/HTML file.

```
top.AddXML = addXML;
top.AddSVG = addSVG;
top.RemoveSVG = removeSVG;
top.RemoveXML = removeXML;
```

Illustration 20: Accessing SVG functions from another document

```
var displayText = "<text x=\""+ ((blockStartPointX - (sectionName.length * 4)) - 4) +"\'
                  y=\""+ (greenLineStartY + 1) +"\' style=\'font-family:palatino; font-size:8\'>
                  "+sectionName+" </text>"
AddSVG(displayText, where);
```

Illustration 21: An example of a string construct to be passed to the SVG document

SVG elements are created 'on the fly' and passed to the embedded SVG document as strings populated by variable parameter values. This technique allows the dynamic scaling of the narrative display and creation of unique graphical curves for the Honey & Mumford display. Note that AddSVG() is an external function call and is executed by the corresponding function in the SVG document script.

Another type of element passed to the SVG document is the path element created for the Honey & Mumford curve. Defining path variable in SVG is difficult due to the counter intuitive nature of the variables used to express complex shapes using the mathematical functions being invoked. Many SVG tutorial texts recommend that path expressions should be defined graphically in an external tool such as Adobe Illustrator, and imported to the SVG document. In the case of the narrative analyser this is not feasible, as individual curves must be generated for the Honey & Mumford display.

```
var mentebarCurve = "<path  fill=\'none\' stroke=\'#000000\' stroke-width=\'4\'
                     stroke-miterlimit=\'2\' d= \'M "+ activistX +" 50
                     T" + reflectorX +" 110  T "+ theoristX +" 170 T "+ pragmatistX +" 230\'>"
AddSVG(mentebarCurve, "Mentebar Curve");
```

Illustration 22: An example of a Path statement to be passed to the SVG document

```
<g id="Mentebar Line">
    <text x="10" y="15" style="font-family:palatino; font-size:10"> Honey and Mumford </text>"
    <text x="43" y="30" style="font-family:palatino; font-size:10"> Values </text>"
    <text x="10" y="280" style="font-family:palatino; font-size:10"> Activist: </text>"
    <text x="10" y="295" style="font-family:palatino; font-size:10"> Reflector: </text>"
    <text x="10" y="310" style="font-family:palatino; font-size:10"> Theorist: </text>"
    <text x="10" y="325" style="font-family:palatino; font-size:10"> Pragmatist: </text>"
    <line style="fill:none; stroke:grey; stroke-width:1; stroke-dasharray:2,3" x1="62" y1="51"
          x2="62" y2="229"/>
    <text x="30" y="243" style="font-family:palatino; font-size:10">0</text>"
    <text x="56" y="243" style="font-family:palatino; font-size:10">20</text>"
    <text x="86" y="243" style="font-family:palatino; font-size:10">40</text>"
</g>
```

Illustration 23: An example of an SVG background element

Background elements of the narrative analyser display are defined in the SVG document. Dynamic elements, which are generated at run time, have root elements defined in the SVG document.

3.5 Key functionality in the JavaScript code

```
// Scale the display
var displayHeight = initialDisplayHeight;
var displayWidth = initialDisplayWidth;
var blockStartPointX = initialBlockStartPointX;
var blockStartPointY = initialBlockStartPointY;
var learnerEvents = logFile.getElementsByTagName(learnerName);
var numberOfLogFileEvents = learnerEvents.length;
var narrativeSections = courseNarrativeFile.getElementsByTagName("section");
var numberOfSections = narrativeSections.length;
var narrativeSubsections = courseNarrativeFile.getElementsByTagName("subsection");
var numberOfSubsections = narrativeSubsections.length;
var numberOfLogNarrativeJumpEvents = (numberOfLogFileEvents - numberOfSubsections);
if (numberOfLogNarrativeJumpEvents < 0)
    numberOfLogNarrativeJumpEvents = (numberOfSubsections - numberOfLogFileEvents);
var blockLength = Math.round(displayWidth / (numberOfSubsections + numberOfLogNarrativeJumpEvents));
var blockHeight = Math.round(displayHeight / ((numberOfSubsections + numberOfLogNarrativeJumpEvents) * 2));
var initialYLineLength = blockHeight;
var initialXLineLength = blockLength;
var greenLineStartX = (blockStartPointX + blockLength);
var greenLineStartY = (blockStartPointY + blockHeight);
var lineStartX = (blockStartPointX + blockLength);
var lineStartY = (blockStartPointY + blockHeight);
var xLineLength = initialXLineLength;
```

Illustration 24: Scaling the log narrative dislpay

Scaling of the access log narrative display with reference to the course narrative display is accomplished by defining all key dimensioning variables at run time relative to the total numbers of narrative events in the course narrative file and the number of narrative jump events derived from the access log file for the selected learner. Key dimensioning variables include:

- blockHeight of each narrative element block in the narrative display.
- blockWidth of each narrative element block in the narrative display.

- YlineLenght is continuously computed based on the relative offset of narrative jumps or set to a default defined when the narrative display is dimensioned.
- XLineLength is normally set to a default defined when the narrative display is dimensioned.

3.5.1 Core JavaScript and SVG functionality

The four core functions that generate the individual display sections are:

- **makeNarrativeEvents()** This function builds a course narrative display by:

  ■Extracting the narrative element sequence from the course narrative file.

  ■Dimensioning the display elements based on the number of narrative elements found in the course narrative file for the selected learner.

  ■Defining Section and Subsection labels and their display coordinates.

  ■Updating the SVG DOM maintained by the SVG document.

- **makeLogFileEvents()** This function builds a correlated course narrative display by:

  ■Extracting narrative events relating to the selected learner from the access log file.

  ■Calculating the narrative offsets of individual access log events relative to the corresponding narrative elements in the course narrative file.

  ■Dimensioning the display elements based on the number of narrative elements found in the course narrative file for the selected learner combined with the number of narrative jump events found by correlating the access log events with the course narrative.

  ■Generating narrative jump lines and narrative termination event displays.

  ■Defining subsection labels and their display coordinates according to the options selected in the *Select Name Display* pull down menu.

  ■ Updating the SVG DOM maintained by the SVG document.

- **makeTimeLine() and makeTimeLineEvent()** These two functions populate the time line display area when the correlated course narrative option is selected in the *Select Narrative Display* pull down menu by:

  ■Extracting narrative events relating to the selected learner from the access log file.

■Dynamically scaling the vertical dimension of the time line display according to the maximum number of access log events per day found in the access log.

■Building the event tracing columns of red and green blocks that support analysis of the temporal relationship of the access log events.

■Updating the SVG DOM maintained by the SVG document.

- **CreateMentebarLine()** This function populates the honey & Mumford display area when a learner is selected in the *Select Learner* pull down menu by:

    ■Extracting the Honey & Mumford learning style classification values from the learner model file.

    ■Generating a unique display curve from these values using the SVG <path> statement.

    ■Updating the SVG DOM maintained by the SVG document.

Listings of these four key functions and some others can be viewed in Appendix 3, C*ore functionality code listings.*

3.5.2 Supplementary code functionality

```
// Add markings to the timeline
elementCount += makeTimeline(logFile, learnerName, numberOfLogFileEvents, monthsToDisplay);
```

Illustration 25: An example of the elementCount variable being updated

The **elementCount** variable is central to maintaining an index of the numbers of individual SVG elements that have been added to the SVG DOM by the preceding four core functions. All functions that update the SVG DOM pass return values to the calling function, which define the number of elements they have added to the structure. This variable, incremented by the return values, has an important role to play in the dismantling of the SVG DOM when display configuration changes are required.

```
function addSVG(xml2parse, where)
{
    document.getElementById(where).appendChild(parseXML(xml2parse,document));
}

function removeSVG(where)
{
    oldElement = document.getElementById(where).lastChild;
    oldElement.parentNode.removeChild(oldElement);
}
```

Illustration 26: AddSVG and RemoveSVG functions

The two short functions **AddSVG()** and **RemoveSVG()** execute in the SVG document and are responsible for adding elements to and removing elements from the SVG DOM respectively.

```
function getOffset(courseNarrativeFile, logFileSubsectionName, logFileSectionName, numberOfSubsections )
{
    var logFileSubsectionOffset = 0;
    for (narrativeFileIndex = 0; narrativeFileIndex < numberOfSubsections; narrativeFileIndex++)
    {
        var narrativeSectionName =
        courseNarrativeFile.getElementsByTagName("subsection")[narrativeFileIndex].parentNode.childNodes(0).firstChild.nodeValue;
        var narrativeSubsectionName =
        courseNarrativeFile.getElementsByTagName("subsection")[narrativeFileIndex].childNodes(0).firstChild.nodeValue;

        if ((narrativeSubsectionName == logFileSubsectionName) && (narrativeSectionName == logFileSectionName))
        {
            logFileSubsectionOffset = narrativeFileIndex;
            break;
        }
    }
    return (logFileSubsectionOffset);
}
```

Illustration 27: getOffset() function

The **getOffset()** function returns the offset of the passed access log event relative to the matching element in the course narrative file. This value is used to determine narrative jump distance and direction.

Finally the issue of drawing order in the SVG document should be noted. As SVG graphics are displayed strictly according to the order their definitions appear in the SVG document, care must be taken to avoid occlusion or partial hiding of display elements by elements declared later in the document.

3.6 Summary

Chapter three has reviewed the software implementation designed to support the prototype narrative analyser, employing software technologies including HTML, JavaScript, and Scalable Vector Graphics, chosen based on the discussion presented in section 2.5, *Implementation Technologies*, of chapter two, *The State of the Art.* A visualisation design and implementation can now be completed in accordance with the achievement of objective two defined in chapter one, *Introduction*, which is - "To design and implement a prototype narrative analyser".

The competed prototype narrative analyser will be utilised as the focus of chapter five, *Trial and Evaluation*, in which the research questions driving this dissertation will be addressed. Key features of the prototype narrative analyser software design include:

1. The requirements placed on the browser are simply that it should support JavaScript and have an SVG display capability – the Microsoft Internet Explorer web browser currently ships with the Adobe SVG plug-in which was used as the foundation for the prototype narrative analyser. This form of the narrative analyser is ideally suited to the role of realising a visualisation accessible to learners, while a Java based version featuring file I/O functionality may be better suited to the potential role for narrative analysis as a component in the suite of course composition tools.
2. All narrative dimensioning variables are dynamically updated to allow scaling of the course narrative and correlated course narrative displays.
3. Live data can be incorporated with relative ease, the key requirement being the implementation of functionality to translate, parse, and format the Apache access log files for analysis and display.

4. Visualisation design.

4.1 Introduction

Chapter four will review the visualisation design implemented for the prototype narrative analyser, building on the techniques researched in chapter two, *The State of the Art*, and the software implementation documented in chapter three, *Software Implementation*. The visualisation design and implementation is motivated by the achievement of objective two defined in chapter one, *Introduction*, which is - “To design and implement a prototype narrative analyser”. In chapter five, *Trial and Evaluation,* the prototype narrative analyser will be the subject of user testing and the focus of discussion in which the research question driving this dissertation can be addressed.

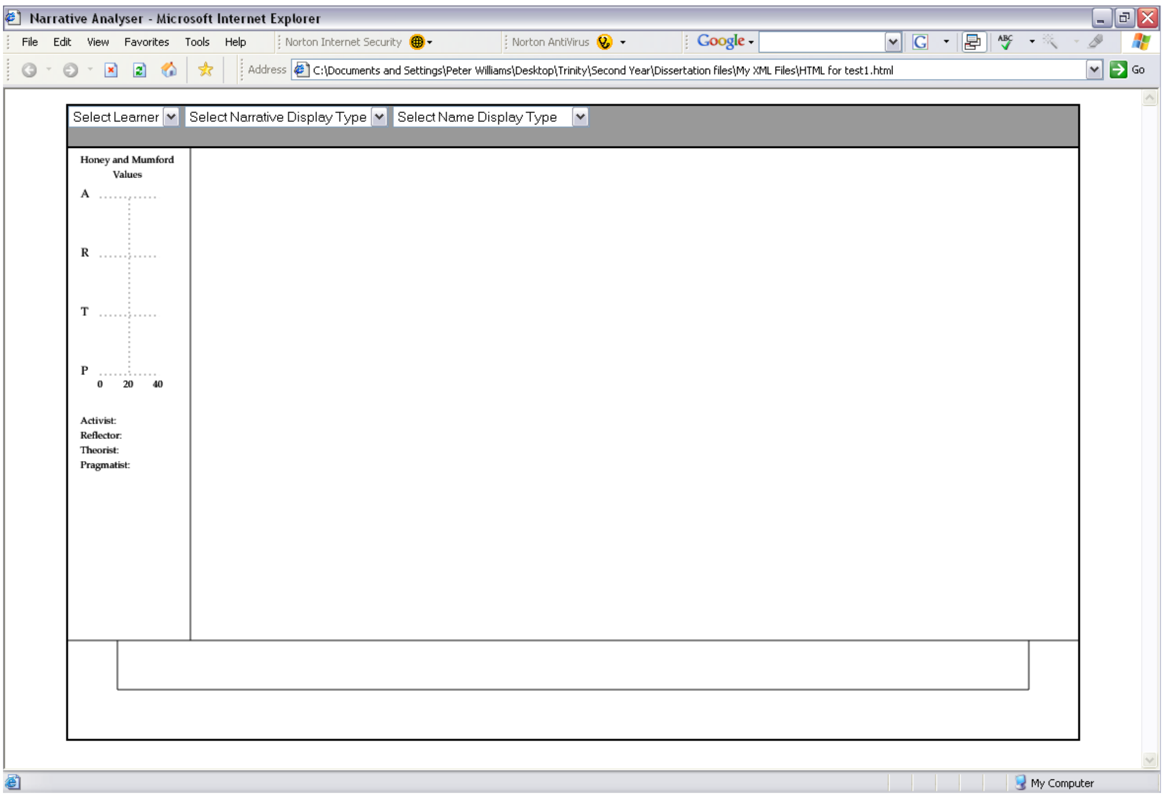


Illustration 28: The prototype narrative analyser in Internet Explorer

The design objectives stated in chapter one, *Introduction*, require the visualisation design to be 'simple, clear, and intuitive'. To this end, software support for the visualisation has been implemented, as seen in chapter three, *Software Implementation*. The prototype narrative analyser is a browser based web application and no software downloads or installations are required in order to access it. All functionality is delivered from the core web page.

The visualisation design detailed in this chapter aims to avoid 'cognitive overload' and 'lost in hyperspace' effects [Conklin, 1987] by making use of clear and simple representation devoid of unnecessary graphical clutter and distractions. Screen grabs of the individual correlated course narrative displays, for all four simulated learners, can be viewed in appendix 2, *Learner correlated narrative displays*.

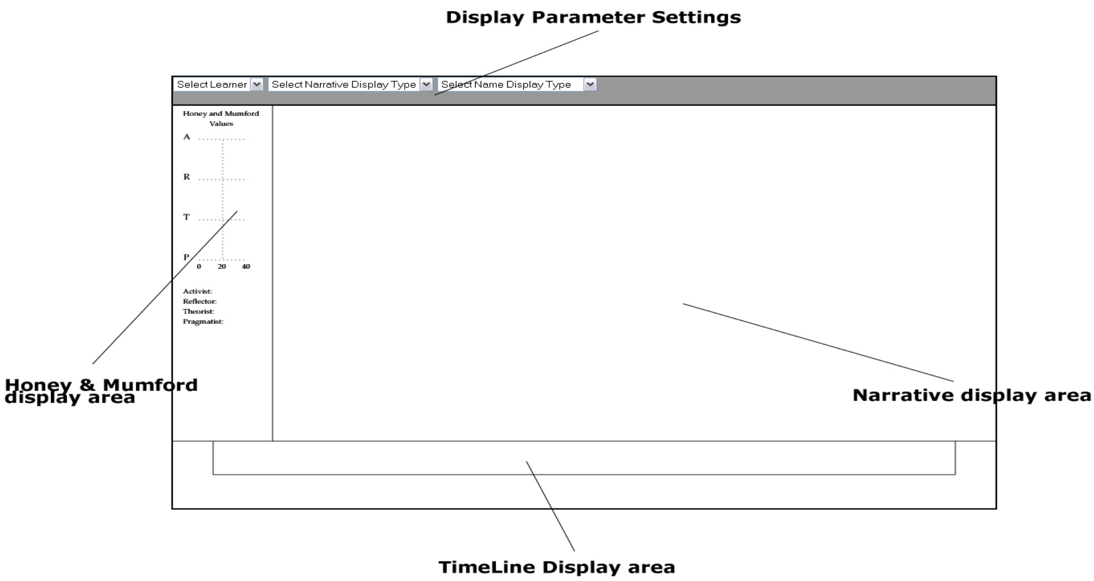


Illustration 29: Prototype narrative analyser display areas

4.2 Visual display areas

The prototype narrative analyser display is divided into four visually separate areas. Each has unique properties when different narrative or learner display options are selected. The display parameter settings area consists of a number of pull down menus, which allow selection of learners and narrative display types, and options for the display of subsection names when displaying access log narratives. The Honey & Mumford display area features a display of the learners' Honey & Mumford learning style classification in both numerical and graphical form.

The Narrative display area is where the various permutations of the course and access log file based narrative structures are displayed. This display is software configured for optimal scaling of the displayed narrative structure to the available display area by adjusting narrative subsection block sizes based on the numbers of subsections in the course narrative and the number of narrative jump events found in the access log file. Narratives are drawn from the top left corner of the display area and extend by the width of one narrative subsection block to the right for each course narrative event or log file access narrative event.

The Time Line display area shows the access log history based on the number of accesses by day. This display is scaled vertically based on the maximum number of accesses on any one day and horizontally based on the number of days included in the access log file being examined. The visual link between the time line and the narrative displays is deliberately broken. This separation reinforces the distinction between the logical time of the narrative display, whereby individual narrative subsection blocks are consistently sized relative to each other and to the nominal left to right direction of time flow, and the absolute distribution of the access history events against time elapsed as shown in the time line display.

4.2.1 The Honey & Mumford display

The Honey & Mumford graph display is designed to convey an 'at a glance' impression of the

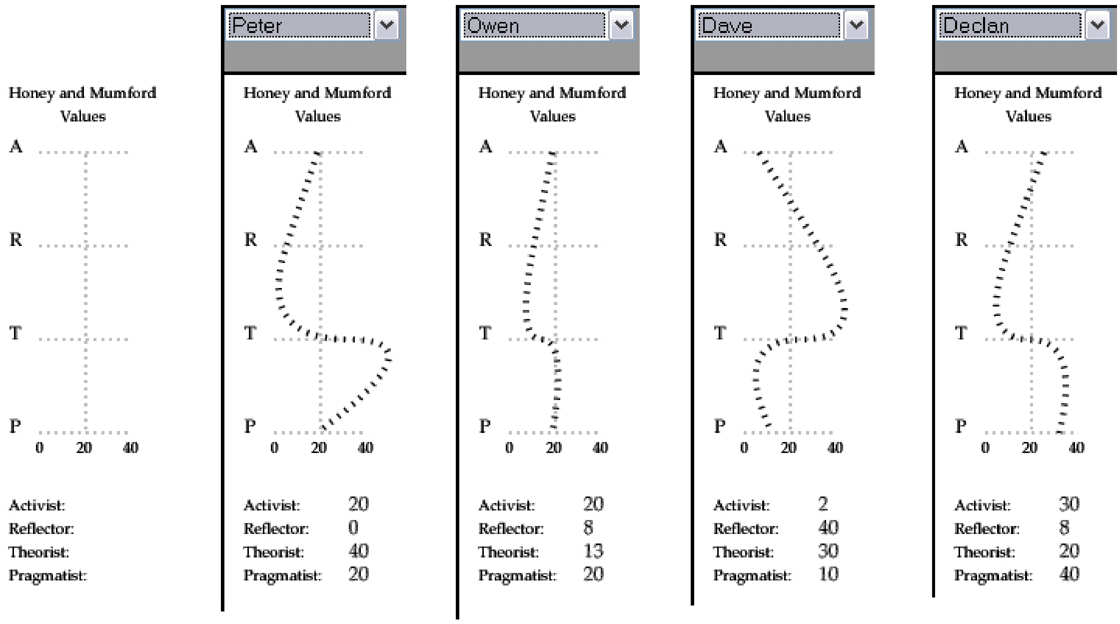


Illustration 30: Honey & Mumford graph and discrete value displays

results given by the learner to the Honey & Mumford learning style questionnaire completed before creation of the suggested course narrative by the adaptive engine. Learners with dominant characteristics in any of the four Honey & Mumford learner types will have distinctive curve shapes biased towards the dominant characteristic. This feature allows for rapid assimilation of key learner characteristics which, when combined with the relevant narrative displays, aims to provide the learner or designer with the basis of meta cognition by contrasting their self declared learning style with the reality of their interactions with the adaptive e-Learning system. Discrete values are also displayed to support more in-depth analysis of the Honey & Mumford learning style characteristics.

4.3 The course narrative display

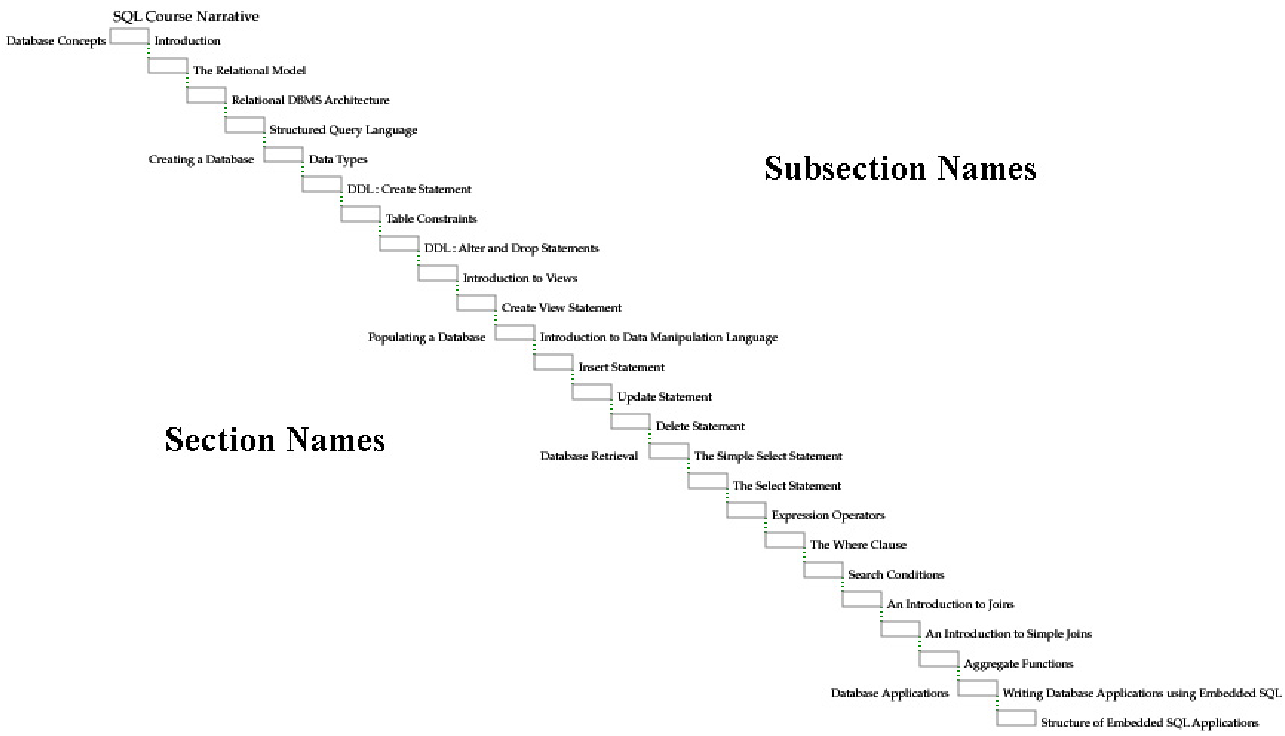


Illustration 31: A course narrative display

The course narrative display is constructed at run time based on the personalised course structure file output by the adaptive e-Learning system. The course structure is created from the pool of available learning objects with reference to the learners' learning style and declared prior knowledge. Narratives are displayed as a staircase extending from the top left of the narrative display area towards the bottom right hand corner of the narrative display area. Thus the first narrative event is found at the top left and the last narrative event is nearest to the bottom right hand corner of the narrative display area. As this is the original sequence of events with which the access log narrative events will later be correlated the staircase will be unbroken. This can be described as the nominal sequence and forms a linear narrative structure.

Illustration 32: A single narrative eventt

The concept of time applied in the narrative display is based on the logical sequence of narrative events relative to each other rather than the relative positions of the narrative events with respect to the passage of real time. The relationship between narrative events is based on '*A happens before B*' with each narrative event offset in the Y axis by a display offset value based on a run time calculation of the available display area relative to the number of narrative events contained in the course narrative file.

The decision to employ a consistent narrative event block size based on the size of the available display area in logical sequence rather than variable narrative event block sizes which are displayed relative to real time is intended to reduce the number of variable elements displayed and thus reduce the possibility of causing 'cognitive overload' [ref]. Furthermore, the use of a consistent narrative event block size allows meaning to be attached to the size used. In the case of the course narrative display the narrative event block size is directly proportional to the length of the course narrative. Thus longer course narratives will result in smaller narrative event block sizes, building an important link between visual scale and narrative length. This relationship between visual scale and narrative length gains deeper significance if user selectable display regions are implemented whereby only a portion of the complete narrative structure may be visible on screen.

Section and subsection names are displayed in the course narrative display at all times, irrespective of the setting of the 'name display' menu. This is because the structure of the course narrative display will always be in the form of an unbroken staircase as all narrative events are in logical order. Section names are shown to the left of their first subsection narrative event and subsection names are displayed to the right of each narrative event.

Thus the important information that can be gleaned from the course narrative display will be encoded in the block size of each narrative event, combined with the number of sections and subsections shown. As the number of subsections and subsections included in the course narrative are dependent on the adaptivity applied by the adaptive engine component of the adaptive e-Learning system this course narrative display conveys a cognitive overview of the suggested course narrative. When combined with the Honey & Mumford display for the learner this can form the basis for an intuitive expectation of the emergence of a characteristic pattern of accesses based on the learner type. This is the foundation on which course designers and administrators can monitor or validate important aspects of the performance of their courses. Individual learners can use the course narrative display in the same way to aid understanding of their individual learning style, thus building a foundation for the development of meta cognitive insight.

4.4 The correlated course narrative display.

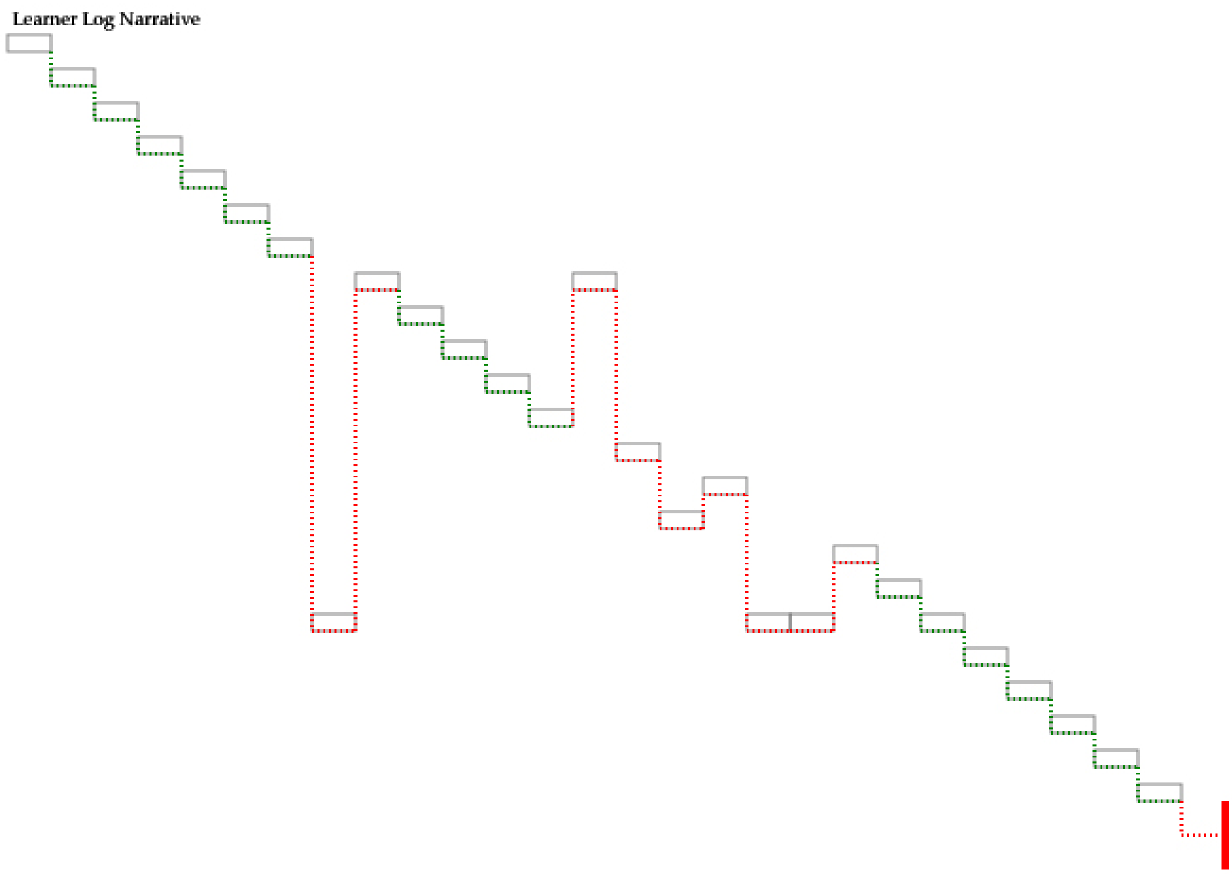


Illustration 33: A correlated access log display

The correlated access log narrative display is constructed at run time based on the personalised course structure file output by the personalised e-Learning system combined with the learner's access history as gleaned from the access log file generated by the server hosting the personalised e-Learning system. Narrative subsections that were repeatedly visited or were accessed out of sequence with the course narrative are clearly visible.

The narrative structure is based on the sequence of narrative events as encountered in the access log file. Narratives are displayed as a staircase of narrative event blocks extending from the top left of the narrative display area towards the bottom right hand corner of the narrative display area. As with the course narrative display the first narrative event is found at the top left and the last narrative event is nearest to the bottom right hand corner of the narrative display area. Narrative jump events are visualised as 'long jumps' from the last narrative event displayed to a vertical position that corresponds to the narrative offset of the corresponding narrative event in the associated course narrative. Narrative jump events are displayed using red dotted lines that continue to the point where the narrative has rejoined the nominal sequence as defined by the course narrative file. Jumps forwards relative to the course narrative might be interpreted as exploration and jumps backwards relative to the course narrative as revision. Maintaining consistency in the structure of the two types of narrative display is an important aid to developing an intuitive understanding of the correlated access log narrative.

As seen with the course narrative display previously, a decision was taken to use a consistent narrative event block size based on the size of the available display area in logical sequence rather than variable narrative event block sizes which are displayed relative to real time. The use of a consistent narrative event block size allows a meaning to be implied by the size used that is valid over the entire narrative length. In the case of the course narrative display the narrative event block size is directly proportional to the length of the course narrative. For the correlated access log narrative the block size is computed from the number of course narrative events and the number of narrative jump events found in the access log file. Thus longer course narratives with many narrative jump events will result in smaller narrative event block sizes, building an important link between visual scale and narrative length and complexity. This can benefit learners and designers alike by reducing the tendency towards 'cognitive overload' . Furthermore, this relationship between visual scale and narrative length gains deeper significance if user selectable display regions are implemented whereby only a portion of the complete narrative structure may be visible on screen.

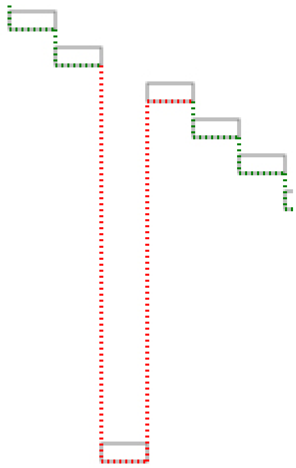


Illustration 34: An example of a forward narrative jump - exploration

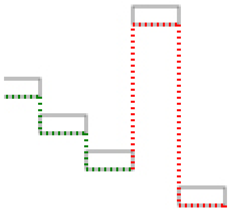


Illustration 35: An example of a backward narrative jump - revision

Forward and backwards narrative jumps are the key transitions visualised which clarify deviation from the course narrative generated by the personalised e-Learning system. Forward narrative jumps indicate the learner is looking ahead to course narrative subsections out of sequence with the course narrative generated by the personalised e-Learning system. The correlated access log narrative has now become a non-linear narrative. There can be many reasons for this, including exploration of the material ahead in order to gauge the difficulty level, or exploration aimed at building an overview of the material. Similarly, backward narrative jumps can indicate the learner is looking back at previously accessed course narrative subsections and may be doing some revision, although it is possible that a backward narrative jump that is not motivated by revision could result from a situation where the learner began the course by jumping forward in the narrative structure but found it necessary to later go back and cover the skipped material.

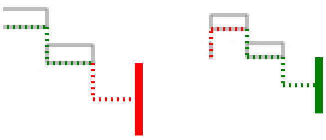


Illustration 36: Examples of Narrative break and Narrative completion

Narrative break events are displayed when the last narrative event encountered in the access log file does not correspond with the last narrative event in the course narrative file. This does not imply failure but merely indicates the last point in the course narrative the learner accessed. Narrative break events are displayed as a vertical red bar. Narrative completion events are displayed when the last narrative event encountered in the access log file matches the last narrative event in the course narrative file. Narrative completion events are displayed as a vertical green bar.

4.4.1 Subsection name display options

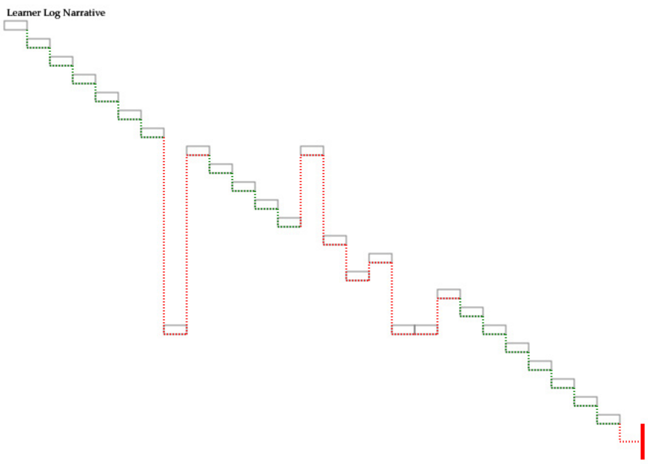


Illustration 37: Access log narrative events displayed with no subsection names

Access log narrative event displays with no subsection names shown facilitate the evaluation of structure without reference to content specifics. Patterns of access that are distinctive may be detected using this display through the comparison of many individual learners' access histories. The gradual development of meta cognitive understanding of these distinctive patterns of accesses could help course designers to optimise the adaptive behaviour of the personalised e-Learning system. It is possible that distinctive patterns of accesses may be correlated with the Honey & Mumford learning styles, providing further insight into the adaptive behaviour and requirements of the personalised e-Learning system as a whole.

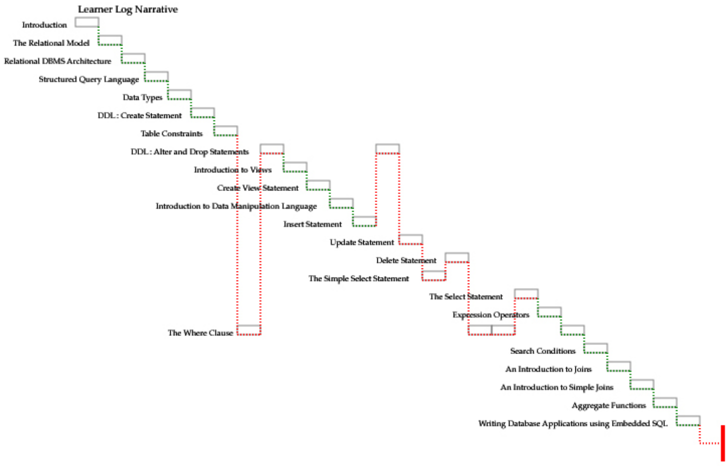


Illustration 38: access log narrative events displayed with all subsection names

Correlated access log narrative event displays with all subsection names shown facilitate the analysis of entire narrative structures as each narrative event can be identified and traced. Narrative subsection names are displayed to the left of the corresponding subsection block in the narrative staircase. Narrative subsection names are displayed on the left in order to preserve the clarity of the narrative staircase structure. Displaying the narrative subsection names to the right of the narrative subsection block would have caused an increase in visual confusion and required users to visually trace backwards across the screen to identify forward jump narrative subsections.

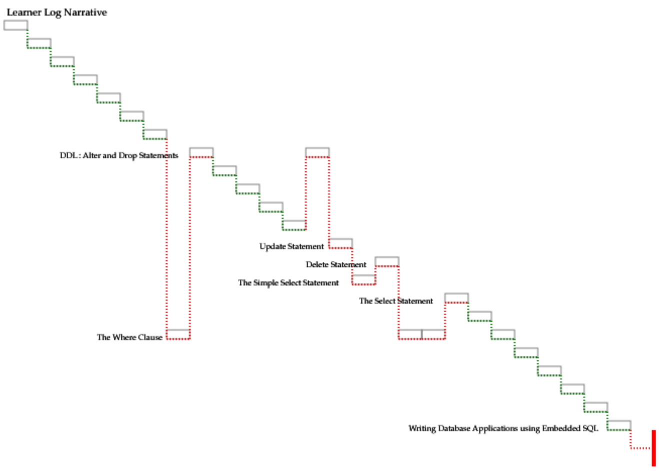


Illustration 39: Access log narrative events displayed with jump and break subsection names

Correlated access log narrative event displays with jump and break narrative subsection names shown highlight only those narrative events which depart from the nominal sequence or linear narrative as displayed for the course narrative. Jump and break narrative subsection names can be used to examine the reasons for narrative jumps and as a technique to help in the diagnosis of comprehension problems with particular learning objects. Narrative subsection names are displayed to the left of the corresponding subsection block in the narrative staircase.

4.5 The time line display

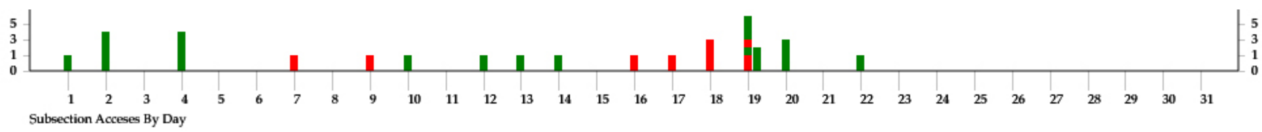


Illustration 40: A time line showing accesses by day

The time line display is created at run time when the correlated access history display is selected. The visual grouping of the narrative analyser display is designed not to offer users a direct correspondence between the narrative display area and the time line. This relates to the decision to employ consistent narrative block sizing in the narrative display area as multiple accesses on the same day would require the block sizes to shrink, obscuring the detailed structure of the narrative.

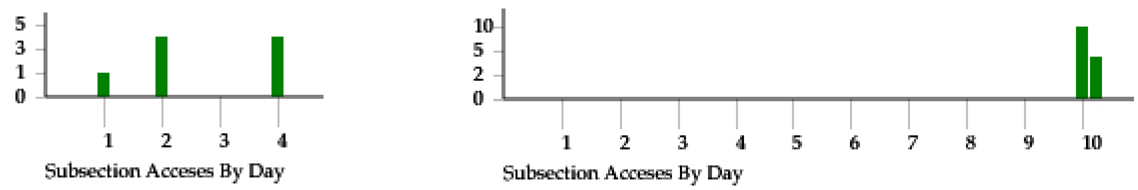


Illustration 41: An example of auto scaling of the 'accesses by day' display

Narrative events are displayed as vertical blocks sized according to the maximum number of accesses by day found for the learner in the access log file. For the prototype narrative analyser scaling of narrative event block height is set to one of two values that equate to column heights of either five or ten blocks. Legends on the time line vertical scale are updated as appropriate to the set column height. Narrative event blocks are stacked by an offset that is half the height of a single access. This stacking algorithm compresses the vertical height of columns while preserving greater clarity for single access events and allowing users to equate column height and number by volume with the number of accesses per day.

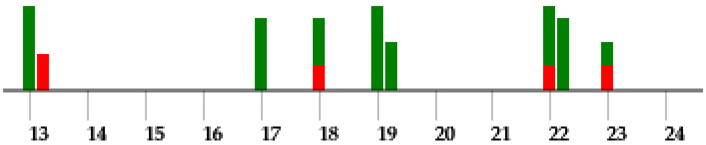


Illustration 42: An example of multiple access events on the same day

Narrative event blocks are displayed in red if they relate to a narrative jump event and green if they relate a nominal sequence or linear narrative event. This colouring scheme is consistent with the colouring employed in the narrative display and reinforces the correlation of events between the real time display of the time line and the logical time display of the narrative display. Detailed analysis of the sequence and timing of narrative events can be done by examination of the columns displayed.

Narrative event columns are associated with a particular day by their proximity to the day marker. If the number of narrative events exceeds the capacity of one column then a new column is started to the right of the first separated by a small distance which reinforces the visual grouping while preserving the relationship with the associated day and enabling detailed examination of the narrative event history.

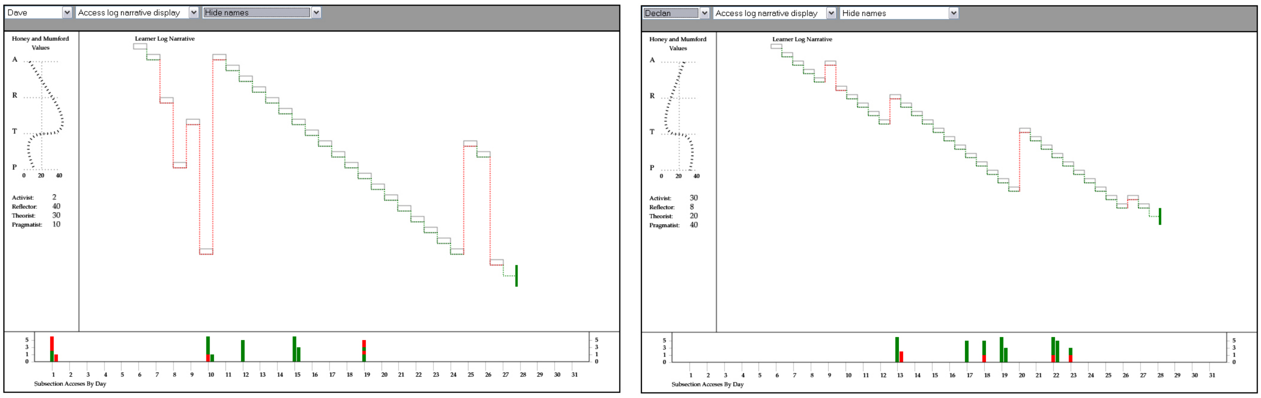


Illustration 43: An example of contrasting access log narratives

One of the most powerful features of the narrative analyser for course designers or administrators is the ability to compare correlated access log narratives or course narratives for different learners. The prototype narrative analyser allows rapid cycling between individual learners, narrative display types, and name display types by selection of the appropriate menu and using the up arrow and down arrow keys to move the selection to the previous or next option. On selection of each option the display is automatically updated, allowing high-level comparisons to be made. The menu structure is split into three categories, grouped as they relate to learner, narrative displays, and name displays.

Illustration 44: Select Learner menu

'Select Learner' allows one of the four test learners to be selected by scrolling to the required learner and selecting by mouse click. Scrolling to any learner name and selecting it by mouse click will cause the Honey & Mumford display area to be updated accordingly. If selections have been made on the 'Select Narrative Display Type' and/or Name Display Type' menus then the narrative display area and the time line will be updated accordingly. Subsequent use of the up arrow and down arrow keys will switch between learners, keeping the selected narrative display type and name display type and allowing fast comparison between learners.

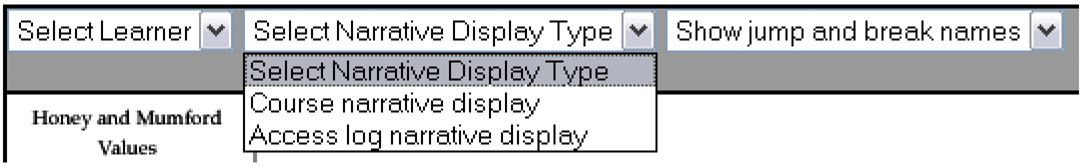


Illustration 45: Select Narrative Display Type menu

'Select Narrative Display Type' allows the selection of one of the two narrative display types available. Scrolling to either 'Course narrative display' or 'Access log narrative display' and selecting by mouse click will cause the narrative display area to be updated accordingly and the time line to be populated or cleared as appropriate. Selecting either narrative display type and using the up arrow and down arrow keys will switch between the two narrative displays.

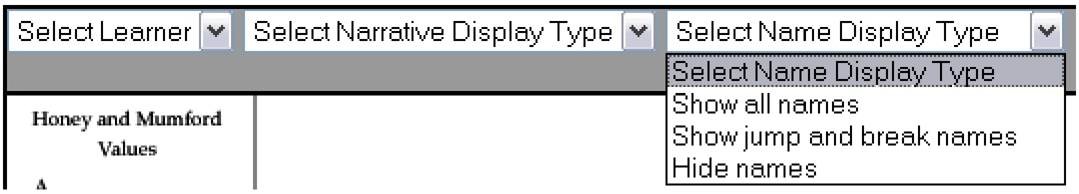


Illustration 46: Select Name Display Type menu

'Select Name Display Type' allows one of the three options for name display to be chosen. Scrolling to 'Show all names', 'Show jump and break names', or 'Hide names' will cause the correlated access log narrative display area to be updated accordingly. Selection of any name display type and subsequent use of the up arrow and down arrow keys will switch the subsection names displayed in the correlated access log narrative display to the chosen mode.

'Select Name Display Type' settings have no impact on the course narrative display as it will always conform to a nominal sequence or linear narrative and therefore the clarity of the narrative structure can not be enhanced by removal of section or subsection narrative event names.

4.6 Summary

Chapter four has reviewed the visualisation design implemented for the prototype narrative analyser, building on the techniques researched in chapter two, *The State of the Art*, and the software design implementation documented in chapter three, *Software Implementation*. The visualisation design and implementation was completed in accordance with the achievement of objective two defined in chapter one, *Introduction*, which is - “To design and implement a prototype narrative analyser”.

The prototype narrative analyser can now be utilised as the focus of chapter five, *Trial and Evaluation*, in which the research questions driving this dissertation can be addressed. Key features of the prototype narrative analyser visual design include:

1. Dynamic scaling of the various correlated narrative displays.
2. A novel display of the Honey & Mumford learning style classification as a curve graph designed to support shape recognition and thus aid rapid assimilation of contrasting learning style definitions and correlated narratives.
3. The decoupled time line display allowing narrative structures to be displayed with constant narrative block size thus clarifying learner access behaviour whilst supporting temporal analysis of events in the correlated narrative display.

5. Trial and Evaluation

5.1 Introduction

As stated in the first chapter the research question posed in this dissertation is based around the exploration of the relationship between adaptive information and its potential visualizations. Specifically, this dissertation asks whether visualisation techniques as applied to the task of visualising correlated course and access history information can provide personalized e-Learning course authors and administrators with an effective analysis and diagnostic tool. In parallel the potential for utilizing such visualization techniques targeted at actual learners has been investigated.

Following from the objectives detailed in chapter 1, T*he State of the Art*, in approaches taken to the design of temporal visualisations was researched. The methodology behind the construction of adaptive e-Learning course narratives by a selection of current e-Learning systems was researched, with a particular focus on the APeLS system developed at Trinity College, Dublin. The Honey & Mumford system of determining a student's learning style preferences, implemented on the APeLS system, was investigated. Potential implementation technologies that satisfy the requirement of providing access to the visualisation for web-based learners without the complexity of application installation were evaluated.

The knowledge gained from the background research phase was applied to the design and implementation of a prototype narrative analyser intended for use as part of an evaluation of the potential uses of a visualisation of correlated course and access history information. A set of evaluation interviews were conducted with three e-Learning course designers based at Trinity College with the aim of collecting qualitative and anecdotal evidence so that the research question could be answered. Each interviewee was first given an introduction to the narrative analyser, it's key functionality, and features. The introduction consisted of a short slide show that highlighted the visual features of the prototype design and the data sources used to compile the visualisation, followed by 'hands on' familiarisation.

Responses to individual questions were not the sole focus of the interviews. The structure of the series of questions was designed to prompt a wider ranging discussion that could evaluate the prototype narrative analyser in the context of not just the artefact presented but of the potential for the visualisation techniques employed with the aim of answering the research question.

The evaluation interviews consisted of two distinct sections. The first section explored the effectiveness of the prototype visualisation by posing a series of nine questions, such as “Did learner Dave complete the suggested course?”, or “Which learner employs revision extensively as part of their learning style?”. These questions are directed at determining the ability of the visualisation to:

1.Encapsulate knowledge about learner behaviour patterns and highlight key events in the correlated course narrative and thereby assist course designers in the validation of (or detection of problematic elements with) course narratives.

2. Illustrate the correlation between learning style models and real world behaviour patterns.

Answers to the first series of questions were determined by 'hands on' use of the prototype narrative analyser. A summary of the responses to and the issues raised by these questions is presented in the *Usage and Interpretation* section of this chapter.

The second section of the interviews followed the form of a series of questions designed to explore the potential application of the narrative visualisation techniques incorporated in the prototype in order to:

1.Provide individual learners with a feedback mechanism that would enable the development of meta cognition with regard to their learning style.

2.Empower course designers with visualisation tools intended not only for post hoc analysis of access logs but as part of the suite of design time tools.

This second set of questions addressed topics such as, “Do you think that developing meta cognition with regard to their individual learning style would help learners to approach learning differently or with greater insight?”, and “As a course designer, would filtering the set of learners being analysed based on learning style assist you to detect trends in learner behaviour and thus to modify course elements or structure? ". Answers to the second series of questions were determined through discussion and reference to the prototype narrative analyser. A summary of the responses to and the issues raised by these questions is presented in the *Applications of Narrative Visualisation* section of this chapter.

This chapter concludes with a review of the results of the user evaluation of the prototype narrative analyser. A full list of the questions posed during the interviews, in the order they were posed, is supplied in Appendix 1 *Evaluation Structure and Questions.* A grid providing a summary of the responses and issues raised during the evaluation interviews is presented in Appendix 4 *Evaluation Responses.* Selected screen grabs taken of the prototype narrative analyser displaying correlated course narrative results are supplied in Appendix 2 *Learner Correlated Narrative Displays* that can be used in conjunction with this chapter.

5.2 Usage and Interpretation

The questions posed in section one of the evaluation are posed in an interleaved manner but naturally fall into two categories. These categories are discrete event location and abstract interpretation.

5.2.1 Discrete event location

Questions of this nature posed relate to discrete events identified in the visualisation. Answers to these questions are determined by navigating the user interface of the prototype narrative analyser and directly interpreting the visualisation. The two specific questions of this type posed were:

- Did learner "Dave" complete the suggested course? And
- Which learner employs revision extensively as part of their learning style?

Answers given to the first question about narrative completion showed a clear perception of the green colour of the 'Narrative Completion' bar as having a direct relationship to 'success'. All those interviewed could answer it immediately. It should be remembered that, as noted in the design chapter, that cultural differences can result in different interpretations of colour, and therefore the use of colour in must therefore imply a cultural bias to the visualisation. The second question generated some discussion about learner “Dave” and learner “Declan” as their correlated course narratives both exhibit backward jumps generally interpreted as revision. However, it was quickly decided that learner “Declan” was the best answer as he had clearly revised large sections of the course as it progressed while “Dave” had simply been exploring the course before completing it without any major periods of revision. This highlighted the distinction between backward jumps directly related to a preceding forward jump and backward jumps motivated by a desire to revise.

Discussion prompted by these questions revealed that all those interviewed share the view that the application of visual data mining techniques to access logs could provide a new perspective on the way real world learners interact with the adaptive e-Learning environment. The discussions about the contrasting correlated course narratives for “Dave” and “Declan” revealed how a surprising insight into more than simply the literal history of 'what happened when' could be developed by enabling designers to infer some aspects of the motivation behind the access patterns of individual learners. It was noted that seeing the visual contrast between the correlated course narratives of multiple users fostered this insight. This initiated a discussion topic, addressed further in the next section, about the aggregation of information derived from the correlated course narratives for larger groups of learners.

One interviewee suggested that the availability of a tool such as the narrative analyser showed clearly that there are many questions course designers could pose which would extract valuable insights from the access logs, resource which had not hitherto been exploited.

5.2.2 Abstract interpretation

Questions of this nature posed require interpretation of the visualisation in an abstract manner. They are formulated to investigate the issue of the whether visualisations which simply present information in an accessible form can form the basis for the development of higher-level insight into learner behaviour. Answers to these questions were determined by navigating the user interface of the prototype narrative analyser followed by discussion of the interpretation of the visualisation. Questions that require interpretation of the visualisation in an abstract manner included:

- What does the incomplete narrative of learner “Peter” suggest to you? And
- What does user “Declan”'s log narrative time line tell us about his learning style as contrasted with user “Peter”'s log narrative time line.
- What might user “Owen”'s log narrative suggest, if anything?
- Describe learner “Dave”'s learning style with reference to his log narrative and Honey & Mumford classification?

Responses to these kinds of questions were wide ranging, reflecting the subjective characteristics of the many possible answers to some of them. This subjectivity itself gave rise to the comment that the narrative analysis highlighted behaviour that prompted questions that might not otherwise be asked, such as learner Owen’s early termination of the narrative. It was suggested that this type of event might prompt an email from the administrators to determine if he was OK, or that another level of correlation taking into account attendance records could be justified.

Interestingly, two of the three interviews conducted gave rise to attempts to categorise the Honey & Mumford curve display by relating them to shapes such as waves and noses. On first being asked to examine an individual learner's learning style one interviewee started by examining the discrete values for the Honey & Mumford curve rather than evaluating the curve displayed. This prompted some discussion of the level of experience that might be required to become proficient in using the narrative analyser. The comment “the learning curve for the learning curve!” was recorded. The nature of forward and backward jumps was described as visually reminding one interviewee of stalagmites and stalactites. Suggestions of possible answers became more plentiful as the questions progressed for all interviewees, suggesting that the 'learning curve' is not very steep for those with a pre-existing awareness of the issues being addressed by the visualisation.

Questions which asked for the narrative analysis to be interpreted by reference to learners' defined learning styles uncovered the issue that individual subsections of the course narrative should be themselves categorised according to the 'conceptual load' they carry i.e. how many new concepts are introduced in this subsection?. This factor could have a direct correlation with the resulting access pattern if learning new concepts prompts revision or exploration. It was, however, pointed out as being very positive that the exposure of this issue was the result of using the narrative analyser as it stands, and that this knowledge could inform designers about the possibilities for subsections with a heavy 'cognitive load' to be problematic. The possible inclusion of meta data describing the 'conceptual load' of a subsection in the course narrative file was discussed.

The comment from one interviewee, “teachers like content not process”, prompted the assertion that one valuable spin-off deriving from the inclusion of a narrative analyser in a course development tool kit would be to focus the attention of course designers not just on the content being prepared but on the process by which learners interact with it.

All three interviews featured some discussion of the match between the learner model and the correlated course narrative. The degree to which the two match is not known. To determine a higher-level correlation like this would, it was suggested, involve the definition of meaningful statistics derived from a significant number of learner access histories by reference to their defined learning styles.

The two questions posed concerning improvements to the utility and clarity of the narrative analysis showed a degree of enthusiasm for the concept of visualisation as a course development tool. This points clearly towards a positive response to the initial research question posed in this dissertation as the prototype narrative analyser has clearly demonstrated the potential for visualisation techniques to be of assistance to course designers and administrators.

Suggested improvements discussed included:

- The incorporation of a visual indication of when a learner has redefined their learning style, possibly combined with some method by which it could display the changes to the subsequent portion of the course narrative.
- The development of visual display structures which allow multiple learner or narrative elements to be selected and displayed simultaneously. Selection of learners with similar learning styles by drawing a curve in the Honey & Mumford display area

showed that the initially strange curve display had been accepted as a useful graphical depiction of the learner's learning style.

- Continuation of the remainder of the course narrative after a narrative termination, possibly in a deprecated form such as being greyed out in order to allow users to quickly establish the position in the course narrative the break occurred. Currently, finding this position requires the selection of the uncorrelated course narrative display, while remembering the last subsection accessed by the learner. This option would also satisfy the request for some form of continuous display of the uncorrelated course narrative.
- Implementing a fish-eye view option that allows dense groupings of narrative jump events to be examined. This might be implemented in conjunction with a selectable region of the time line.
- The definition of extra colour coding when multiple learner data is being displayed.
- Implementing a method for the display of access patterns relating to the finer grained learning objects contained in individual subsections of the course narrative. Allied to this point, some form of feedback for course designers relating to the amount of time the learner actually spent looking at the screen relating to an individual learning object would add an extra dimension to the analysis. In a test scenario this might be accomplished using computer vision techniques to determine gaze direction, together with information about which window is active on screen.

All interviewees were positive about the uncluttered appearance of the narrative analysis display and were keen that it should not become visually overcomplicated and thus partly negate the benefits of the visualisation.

5.3 Applications of Narrative Visualisation

As mentioned earlier, the second section of the interviews followed the form of a series of questions designed to explore the potential application of the narrative visualisation techniques incorporated in the prototype in order to:

- Provide individual learners with a feedback mechanism that would enable the development of meta cognition with regard to their learning style. These questions relate to the second part of the research question.

- Empower course designers with visualisation tools intended not only for post hoc analysis of access logs but as part of the suite of design time tools.

The questions again fall into two distinct categories – those that are learner centric and those that focus on the potential for narrative analysis as a component in the suite of course development tools available to course designers.

5.3.1 Learner centric questions

These questions concern the learner and the potential for them to develop meta cognition regarding their learning style. Questions posed included "Do you think learners could benefit from access to visualisations of their own approach to learning?" and "Do you think that developing meta cognition with regard to their individual learning style would help learners to approach learning differently or with greater insight?"

The answers uncovered some interesting observations. Perhaps the most important is the question posed by one interviewee that asked, "If learners know their access pattern is going to be analysed would that change the way they access?" No clear answer to this question can be determined at this stage of development of the narrative analyser as to do so would again involve the aggregation of statistical information as discussed earlier. However, it was felt that in general individual learners would ignore the potential surveillance of their use of the e-Learning system, and that furthermore they might expect the system to be deriving information from learner's experiences. One comment stated, "if there is no implication for exam results then they won't care about it!"

Another observation concerned the potential for some learners to become 'expert learners', focussed on the process of learning rather than the course material, which led to the comment that "the whole world is not going to become more theoretical just because you are". To balance this possibility one interviewee made the observation that "Insight is never wasted. We inform our future decisions based on insight into our past successes and failures. In this case it might be that a learner would change the way he tackled the next course based on insight gained from the analysis of this one, or modify his learning style to better reflect his preferences during this one".

The possibility for learners to develop meta cognition with regard to their learning style could benefit course designers by enabling learners to provide more accurate or insightful feedback regarding the course they are currently pursuing or have recently completed. Also mentioned was the possible improvement in the accuracy of learning style definition due to the feedback provided by the narrative analysis. The might counter the possible tendency for learners to define their learning style according to how they would like to be perceived rather than how they really approach learning, which was illustrated by the comment that "learners with a business background might give rise to the definition of many pragmatic activists".

No negative comments were recorded other than the possible issue of the learner being distracted from the learning goal, and the previously mentioned possibility of the creation of 'expert learners'. Finally, the observation that "Some learners might access it [the narrative analysis] simply because it is there and gain some insights almost by accident, but that is good too..." indicates the positive feeling about the potential for learners to develop meta cognitive insights about their learning style.

5.3.2 Course development tool questions

These questions explore the potential for visual narrative analysis to be incorporated as a component in the suite of course development tools available to course designers. Questions posed included "As a course designer, would filtering the set of learners based on a classification of log narrative styles assist you to detect trends in learner behaviour and thus to modify course elements or structure?" and "As a course designer, would additional functionality, such as enabling changes to content dependencies to be implemented directly from the visualisation, enhance the utility of the narrative analyser significantly?" These questions are not intended to form a comprehensive investigation of the development paths that may be available for the narrative analyser but are instead intended to provide insights which support an answer to the research questions posed in this dissertation.

An initial comment from one interviewee that "course designers are very much in the dark about how learning styles are actually related to the real world usage of the system" points towards a rich seam of data mining opportunities for narrative analysis. The incorporation of higher-level correlations such as evaluating a single subsection or individual learning object against many learners experience of it was discussed. This discussion further opened the issue of using narrative analysis as a means to validate the adaptivity expressed in personalised course narratives, based on learner activity. Questions such as “how many broken narratives were for strong reflectors?” or “how closely do strong pragmatists follow the suggested narrative?” could be formulated and evaluated in a course development tool version of the narrative analyser.

The difficulty of classifying learners based on their access behaviour was again considered but would be of significant interest to all those interviewed, and was deemed “difficult, but not necessarily impossible” by some. Methods of selection of groups of learners by means of a dedicated scripting language or visual techniques such as clicking and dragging over a group of subsections in a narrative display were considered, as was the selection of narrative events to be analysed based on temporal considerations such as “the two weeks before the exam”.

The idea of the 'conceptual load' of a subsection was raised again and the possibility of the course designer being able to re-sequence problematic concepts directly from the narrative analyser was discussed. One interviewee expressed the view that it would be very difficult to integrate this functionality, and that it would perhaps be better to view the narrative analyser as a tool to be used alongside the existing course composition tools.

The question relating to the dynamic modification of a learners 'learning style' definition based on an analysis of their access history was warmly received but again the discussion centred on the difficulty of defining the statistics on which the classification might be based. It was felt that modification in this way would “at least be based on what people actually did rather than on what they said they did”, and it was noted that this form of systematic modification of learning styles would potentially be more accurate than the 'confidence value' based method employed in the AHA! E-Learning environment. It was also noted that if learners were using the redefinition of their learning style as a form of index to the content available that a loop could be formed as the system attempted to 'correct' the changes.

Finally, the many possibilities for the development of visualisation based narrative analysis techniques prompted the comment that “there are so many ways this tool could develop that you need to be careful to choose the best one”. This statement encapsulates the favourable reaction to the prototype narrative analyser and can only support a positive answer to the both parts of the research question addressed in this dissertation.

5.4 Security and privacy considerations

The prototype narrative analyser consists of a browser based Scalable Vector Graphics application created using JavaScript, HTML, and SVG which executes in the client browser environment using the Adobe SVG viewer plug-in.

5.4.1 Threat Analysis

The issues dealt with here refer to a potential deployment version of the narrative analyser rather than to the prototype. The prototype exists as a standalone system intended to illustrate potential display scenarios and does not access the on-line course system. The deployment environment would involve a web front end and a server(s) for content retrieval. Potential threats identified include:

- Student altering log records. As this is a system that may be incorporated as an element of examinable coursework the incentive for tampering with the logs could exist.

    Probability – Medium. Cost – Low.

- Database vulnerabilities. Servers and databases integral to the system can exhibit their own set of security vulnerabilities. These could result in an attack vector that bypasses the adaptive e-Learning system itself.

    Probability – Medium. Cost – High.

- Student altering another student's log or profile data. Make everyone else's course that bit harder by causing the system to generate a narrative incorrectly matched with his or her requirements/abilities.

    Probability – Medium. Cost – High.

- Data translation vulnerabilities. XSL style sheets could be changed such that false data is injected into the Narrative Analyser.

Probability – Low. Cost – High.

- Denial Of Service and Distributed DOS attacks on the servers and databases. The system is not high enough profile to attract serious hacker interest and the potential for blackmail against educational bodies must be low.

  Probability – Low. Cost – High.

- Insecurely set password defaults throughout the system. As the system is compiled from modules developed by separate teams there is great potential for password mechanisms to be abused – or bypassed completely.

  Probability – High. Cost – High.

- Trojan horse/buffer overruns/stack overflows in the narrative analyser itself. Only one instance of the narrative analyser is affected and data integrity is not affected.

  Probability – Medium. Cost – Low.

- Spoof Analyser attacks. If the system is 'live editing' enabled then this type of attack could cause major damage.

  Probability – Low. Cost – High.

- Spoof systems. A determined student or a competing institution might wish to misdirect the Narrative Analyser in order to cover failures or discredit the system.

  Probability - Low. Cost – Low/High.

- Stale supervisor accounts. Old account logins might become compromised.

  Probability – Medium. Cost – High.

5.4.2 A secure approach to development.

The development phase of the project should consider both system design issues and software development issues. Technologies that could be employed in the system infrastructure to improve security include:

- SSL/TLS for links between the narrative analyser and the Adaptive Engine/On-line system. A more fundamental change to the infrastructure might be to employ IPSEC that would not require any changes to existing modules as files could be transmitted as text.

- The system should ask for a secure login password after the secure links have been established as a secondary measure.
- XML signatures or encryption could be incorporated as well but would require modification of the Adaptive Engine/On-line system.
- SSL client authentication could be employed to reduce the probability of 'Spoof Analyser' attacks on a live editing enabled system.
- Multiple firewalls, preferably sourced from different manufacturers isolating the web facing servers from the content network except for controlled traffic.
- Security considerations should be evaluated and key measures incorporated in the initial design documentation. End users of the system should be consulted at this stage to achieve 'buy-in' to the whole idea of system security.
- Active security monitoring built in to the application. Perhaps some heuristics relating to good (or bad) behaviour could be defined. Suspicious activity should be signalled to the supervisors email account or the narrative analyser if active.
- Privacy protection. Separation of student names and numbers in order to preserve anonymity.
- Regular code reviews – preferably undertaken by a third party. Vulnerability to buffer overruns, stack overflows, and the presence of suspicious code blocks are a major focus here.
- Stress testing. Possibly initiated at module level before integration of the system. Typical tests might try passing excessively large log files or malformed XML files at the system. Known vulnerabilities in the XML parsers and many browsers should be considered although they could only affect one instance of the narrative analyser and not the core system. This would need to be reviewed if the system was intended to allow supervisor level editing of narrative structures, as then the risk is obvious.
- Tiger team testing. The integrated system should be subjected to a series of attacks by a tiger team(s).
- Modify design. Any vulnerability uncovered at either the module or system-testing stages should be addressed immediately and the design documentation updated accordingly.
- Physical code security. Source code should be protected using a secure source control tool and backups kept in secure storage. This can minimise the risks of malicious elements being

added to the code by outside attackers but not for insiders.

5.4.3 Deployment issues

Education of the system installation/maintenance personnel and correct induction of new personnel should be made part of the induction procedures. Particular attention should be given to the prevention of insecure default attacks by detailing the known default values and including some form of verification that these values have been updated and reviewed periodically such as a check list to be completed and included in the system manual.

Documented procedures for creation and deletion of user and supervisory accounts including periodical review and renewal of passwords to minimise the use of stale accounts as an attack vector should be created. Finally, a separate copy of the system should be kept for ongoing testing in an off-line manner. This also facilitates a coherent disaster recovery strategy to be put in place to reinstate the system in case any of the above threats should overcome the defences.

5.5 Summary

Chapter four has presented an analysis of the responses given during a set of user test interviews conducted with three adaptive e-Learning course designers in Trinity College, Dublin, and security and privacy issues related to the development and deployment of a 'live' implementation of a narrative analyser were addressed in section 5.4 of this chapter.

The evaluation interviewees were first introduced to the concept of narrative analysis and then asked to determine the answers to two groups of questions. The first series of interview questions were designed to explore the effectiveness of the prototype narrative analysis visualisation by requiring hands-on use of the prototype in order to find answers. The second series were designed to foster discussion of the potential applications of the narrative analysis techniques explored in this dissertation.

The positive response given to the prototype narrative analyser gave rise to many suggestions for extending the functionality provided. These included:

- Modifications to the correlated course narrative display, including a post termination display of the remaining narrative events, and the incorporation of some indication of when the learner redefined their learning style.

- The development of visualisation techniques that would support the analysis of multiple learner behaviour patterns, and visualisation techniques for the selection of matching criteria.
- The implementation of zooming features to facilitate the inspection of sections of narrative where learner activity was most active.
- The extension of the narrative analysis to the level of individual subsections of the narrative allowing course designers to analyse activity related to individual concepts.

A key issue that emerged during the user interviews was the concern expressed that there was currently no validation available of the match between learning style definitions and actual learner behaviour. The difficulty inherent in the formulation of meaningful statistics by which learner behaviour patterns could be categorised was discussed and deemed difficult, but not dismissed as impossible, highlighting the need for further research in this area.

The idea of the 'conceptual load' of individual narrative subsections was discussed with reference to the possibility of course designers being able to re-sequence problematic concepts directly from the narrative analyser. One interviewee suggested that it would be difficult to integrate this functionality, and that it would perhaps be better to view the narrative analyser as a tool to be used alongside the existing course composition tools. However, broad support was given to the possible implementation of the narrative analysis techniques described in this dissertation to a version of the narrative analyser to be included in the suite of course composition tools available to designers. This relates directly to the primary research goal addressed and clearly supports the assertion that narrative analysis techniques can provide personalised e-Learning course authors and administrators with an effective analysis and diagnosis tool.

A parallel question asked if learners could benefit from the availability of a visualisation of their individual correlated access history, and whether this would support the development of meta-cognition. While concerns about the potential creation of a class of 'expert learners' focussed on optimising their learning style definition were voiced, the consensus formed was that any insight learners derived from the analysis of their behaviour was inherently of value, even if it is developed by accident. One other important concern highlighted was the possible deliberate manipulation of their access behaviour by learners aware that such analysis is possible. Thus, if care is taken to reassure learners of the neutrality of their learning behaviour (i.e. it will not be used in evidence against them!), it is likely that narrative analysis techniques could support the development of meta-cognition.

6. Conclusion

6.1 Introduction

The research question defined in chapter one of this dissertation defined the area of research to be examined as concerning the relationship between adaptive information and its potential visualizations. Specifically, this dissertation asked whether visualisation techniques as applied to the task of visualising correlated course and access history information could provide personalized e-Learning course authors and administrators with an effective analysis and diagnostic tool. In parallel the potential for utilizing such visualization techniques targeted at actual learners was to be investigated, with a focus on the use of visualization to support meta-cognition.

In order to answer the questions defined above the primary project goal was defined as being "to investigate the potential applications of the visualisation of course narrative structures as generated by personalised e-Learning systems correlated with the actual usage histories of individual learners".

To realise this goal the following objectives were defined and achieved:

1. The areas of visualisation and personalised e-Learning systems were researched and documented.
2. A software environment designed to support a prototype narrative visualisation was developed and implemented.
3. A prototype narrative analyser visualisation was designed and implemented.
4. The prototype narrative analyser was evaluated. Evaluation of the prototype was carried out by means of hands on use of the prototype narrative analyser focussed on determining answers to specific questions posed, combined with individual interviews.
5. The potential for further research in this area was discussed and some strategies for development outlined.

This chapter discusses the objectives of this dissertation and how they were achieved, and concludes with a discussion of future work that may build on the results of this dissertation.

6.2 Review of objectives

6.2.1 Objective 1

To research and document:

- The process by which learning style is classified according to the Honey & Mumford system of classification.
- The methodology supporting the definition of course narrative structures in personalised e-Learning systems.
- The state of the art in approaches taken to the development of temporally correlated visualisations.

The first part of section 2.2 in chapter two introduced the Honey & Mumford system of learning style classification, and the second part described the process by which courses narratives are constructed in personalised e-Learning systems. A particular focus was placed on the APeLS (Adaptive Personalised e-Learning Service) system developed by the Knowledge and Data Engineering Group in the Department of Computer Science at Trinity College, Dublin. The use of the Honey & Mumford learning styles classification and meaning of the term 'narrative' in this context were discussed. Also considered were the approaches to narrative construction taken by the AHA! (Adaptive Hypermedia Architecture) system developed in the Department of Computer Science at the Eindhoven University of Technology, Holland and the 3DE (Design Development and Delivery Electronic Environment Educational Multimedia) system developed as part of a research project which formed part of the European Union IST 5th Framework Programme.

Section 2.3 section of chapter two presented a review of the state of the art in the approaches taken to the visualisation of temporally correlated information, supplemented in Section 2.4 by a review of the novel visualisation techniques proposed in the PlaceTime project [Hampson, Williams, 2004].

Section 2.5 provides an overview of the implementation technologies available for the implementation of narrative visualisations that satisfy the 'no user installation needed' requirement stated in the project goals defined in chapter one.

6.2.2 Objective 2

To design and implement a prototype narrative analyser.

Based on the review of the state of the art, the examination of learner modelling and narrative construction techniques, and the overview of implementation technologies detailed in chapter two, it was decided to design and implement a prototype narrative analyser using a combination of HTML, JavaScript, and Scalable Vector Graphics. Details of the software implementation are described in chapter three, and the narrative visualisation design is detailed in chapter four.

6.2.3 Objective 3

To evaluate the prototype narrative analyser by means of hands on testing and user feedback.

Chapter four provides an analysis of the responses given during a set of user test interviews conducted with three adaptive e-Learning course designers in Trinity College, Dublin. Interviewees were first introduced to the concept of narrative analysis and then asked to determine the answers to two groups of questions. The first series of interview questions were designed to explore the effectiveness of the prototype narrative analysis visualisation by requiring hands-on use of the prototype in order to find answers. The second series were designed to foster discussion of the potential applications of the narrative analysis techniques explored in this dissertation.

Together, these two series of questions and the related discussion illustrate the enthusiasm for a narrative analysis component to be included in the suite of course composition tools available to course designers and administrators. Furthermore, while mindful of the reservations expressed concerning the creation 'expert learners' as they are referred to in section 5.3 *Applications of Narrative Visualisation* of chapter five, it was clearly agreed that providing learners with a visualisation of their individual learning behaviour could support the development of meta-cognitive insight with regard to their learning style, with benefits for both learners and course designers alike.

Thus, as demonstrated by the successful evaluation described in chapter five, *Trial and Evaluation*, it can be seen that this dissertation has addressed both parts of the research question defined in chapter one through the achievement of the objectives detailed and goals specified.

6.3 Contribution to the State of the Art

The prototype narrative analyser display augments the state of the art in visualisation design. Key features include:

4.Dynamic scaling of the various correlated narrative displays.

5.A novel display of the Honey & Mumford learning style classification as a curve graph designed to support shape recognition and thus aid rapid assimilation of contrasting learning style definitions and correlated narratives.

6.The decoupled time line display allowing narrative structures to be displayed with constant narrative block size thus clarifying learner access behaviour whilst supporting temporal analysis of events in the correlated narrative display.

6.3 Future work

Chapter five, *Trial and Evaluation,* detailed many possible areas for development of the narrative analyser prototype as evaluated. These included:

- Modifications to the correlated course narrative display, including a post termination display of the remaining narrative events, and the incorporation of some indication of when the learner redefined their learning style.
- The development of visualisation techniques that would support the analysis of multiple learner behaviour patterns, and visualisation techniques for the selection of matching criteria. This might be accomplished by making use of the third display dimension in the narrative display. A combination of SVG and X3D support would be required in the visualiser software.
- The implementation of zooming features to facilitate the inspection of sections of

narrative where learner activity was most active.

- The extension of the narrative analysis to the level of individual subsections of the narrative allowing course designers to analyse activity related to individual concepts.

Other possible improvements noted during the implementation of the prototype narrative analyser included:

- File saving – exporting an updated DOM for file save operations. This is not possible in a JavaScript environment and would require access to Java functionality to implement. Therefore it would be considered as part of a course composition tool kit version of the narrative analyser.
- Colour coding of the Honey & Mumford curve by dominant trait and the inclusion of discrete indicators reflecting the individual Honey & Mumford values on the crossbar elements of the curve display background.
- Incorporating the display of the individual pagelet accessed as a vertical bar registered against narrative blocks as an indication of the offset in the range of available pagelets.
- Display learner model change points as events on the time line and update H&M curve display to the latest version – display earlier definitions by clicking an earlier definition icon in the time line display area.
- Implement warping and zooming of the narrative display based on the time line.
- Develop a method of comparing narrative access histories. A correlated narrative history signature could be defined based on narrative transition events and statistics including:
    - Nominal sequence frequency and order.
    - Forward jump frequency, number, and distance.
    - Backwards jump frequency, number, and distance.
    - Number and frequency of visits to subsection or pagelet.
    - Completeness of narrative traversal.

- Apply the course narrative signature described above to the tasks of contrasting access styles and modifying learning style definitions based on 'learnt' information, and thereby support the analysis of access patterns with a view to the validation of the learner model against learner access behaviour and the adaptivity applied by the personalised e-Learning system.

References

[Abel et al, 2000] Abel, P., Gros, P., Loisel, D., Paris, J.P., Russo, C., Trichaud, N. *Mapping Information onto 3D virtual Worlds* Proceedings of the International Conference on Information Visualization, 2000

[Breiteneder et al, 2002] Breiteneder, C., Eidenberger, H., Fiedler, G., Raab, M. *Lookmark: A 2.5D Web Information Visualization System* Proceedings of the EURASIA-ICT Conference, Tehran, Iran, 2002

[Bruen et al, 2002] Bruen, C., Conlan, O., *Adaptive ICT support for Learning Styles – A Development Framework for re-useable learning resources for different learning styles & requirements* ITTE 2002 annual Conference of the Association of Information Technology for Teacher Education 2002

[Canavan, 2004] Canavan, J. *Personalised e-Learning Through Learning Style Aware Adaptive Systems*, Trinity College, Dublin, 2004

[Carpendale et al, 1997] Carpendale, T., Cowperthwaite, D.J, Fracchia, F.D, Sheelagh, M. (1997) *Extending Distortion Viewing from 2D to 3D*: *IEEE Computer Graphics and Applications,* Volume 17, Issue 4 (July 1997)

[Chen, 2004] Chen, C. *Information Visualization, Beyond The Horizon*, *Second Edition*, Springer-Verlag, London, 2004

[Chen et al, 2005] Chen, C., Geroimenko, V. *Visualizing Information Using SVG and X3D*, Springer-Verlag, London, 2005

[Conklin, 1987] Conklin, J., (1987) *Hypertext: An introduction and survey.* IEEE Computer, 20(9)

[Conlan et al 2002] Conlan, O., Wade, V., Bruen, C., Gargan, M. (2002) *Multi Model, Metadata Driven Approach to Adaptive Hypermedia for Personalised e-Learning* Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Malaga 2002

[Conlan et al, 2003] Conlan, O., Hockemeyer, C., Wade, V., Albert, D. (2003) *Metadata Driven Approaches to Facilitate Adaptivity in Personalised e-Learning Systems* Journal of the Japanese Society for Information and Systems in Education, (2003)

[Dagger et al, 2003] Dagger, D., Wade, V., Conlan, O. *Towards "anytime, anywhere" Learning: The Role and Realisation of Dynamic Terminal Personalisation in Adaptive e-Learning* Ed-Media 2003, World Conference on Educational Multimedia, Hypermedia and Telecommunications, Hawaii, 2003

[Del Corso et al, 2003] Del Corso, D., Ovcin, S., Morrone, G. *A Teacher Friendly environment to foster Reusability and Learner-Centered Customisation in the development of Interactive Educational Packages* Frontiers in Education, 2003

[Dix et al, 2004] Dix, A., Finlay, J., Abowd, G.D., Beale, R. *Human-Computer Interaction, Third Edition*, Pearson Prentice Hall, Harlow, England, 2004

[Geisler, 1998] Geisler, G 1998, *Making Information More Accessible: A Survey of Information Visualization Applications and Techniques,* http://www.ils.unc.edu/~geisg/info/infovis/paper.html, Site Viewed: 16/04/05

[Hampson, Williams, 2004] Hampson, C., Williams, P. *PlacTime report*, Trinity College, Dublin, 2004

[Heller et al, 1998] Heller, D., Plaisant, C., Mushlin, R., Snyder, A., Li, J., Shneiderman, B. *LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records*: proceedings of the American Medical Informatic Association Annual Fall Symposium, Orlando, 1998

[Honey et al, 1992] Honey, P., Mumford, A. (1992) *The Manual Of Learning Styles,* Maidenhead, Berkshire 1992

[Havre et al. 2000] Havre, S., Hetzler, B., Nowell, L. *ThemeRiver: Visualizing Theme Changes over Time* Proceedings of the IEEE Symposium on Information Vizualization, 2000

[Jones & Krishnan, 2005] Krishnan, A., Jones, S., *TimeSpace: activity-based temporal visualisation of personal information spaces* Personal and Ubiquitous Computing, Volume 9, Issue 1, January 2005

[Sarrikoski, 2000] Sarrikoski, L., Salojarvi, S., Del Corso, D., Ovcin, E. (2000) *3DE: An Environment for the Development of Learner Oriented customised Educational Packages* ASEE/IEEE Frontiers in Education Conference, 2000

[Stash et al, 2004] Stash, N., Cristea, A., De Bra, P. *Authoring of Learning Style in Adaptive Hypermedia Problems and Solutions* WWW'04 Education Track, ACM, Ney York, 2004

[Spence, 2001] Spence, R., *Information visualization*, Addison-Wesley, Harlow, England, 2001

[Tufte, 1983] Tufte, E.R., *The visual display of quantitative information*, Graphics Press, Cheshire, Connecticut, 1983

[Tufte, 1990] Tufte, E.R., *Envisioning Information*, Graphics Press, Cheshire, Connecticut, 1990

[Ware, 2000] Ware, C., *Information visualization: perception for design*, Morgan Kaufman San Francisco, 2000

Appendix 1. Evaluation structure and questions

Introduction to the prototype.

1.Introduction to Narrative Analysis.

2.Learner classification by Honey and Mumford values.

3.File types and sources.

4.Correlations between log files and course narrative files.

5.Overview of controls and display properties.

6.Learning style H & M display.

   1.Honey and Mumford value curve.

   2.Honey and Mumford discrete values.

7.Log file and course narrative display.

   1.Nominal sequence ordering.

   2.Narrative jump forwards in the narrative (Exploring).

   3.Narrative jump backwards in the narrative (Revision).

8.Narrative termination.

   1.Completed course narrative.

   2.Incomplete course narrative.

9. Timeline of access history.

   1.Scaling of accesses by day.

   2.Interpretation of jump and nominal sequence accesses.

10.Select Learner pull down menu.

11.Select Narrative display type pull down menu.

   1.Course narrative.

2.Access Log narrative.

12.Select Narrative name display type.

1.Show all names.

2.Show jump and break names.

3.Hide names.

13.Cycling between learners, narrative display types, and name display types.

Usage and interpretation questions.

1.Did learner “Dave” complete the suggested course?

2.Describe learner “Dave”'s learning style with reference to his log narrative and Honey and Mumford classification?

3.Does learner “Declan”'s access pattern suggest a strong Activist/Pragmatist to you?

4.What does user “Peter”'s incomplete narrative suggest to you?

5.Which learner employs revision extensively as part of their learning style?

6.What might user “Owen”'s log narrative suggest, if anything?

7.What does user “Declan”'s log narrative time line tell us about his learning style as contrasted with user “Peter”'s log narrative time line.

8.What would you change about the visualisation in order to improve it's utility?

9.What would you change about the visualisation in order to improve clarity?

Speculative use-case scenario based questions.

1.Do you think learners could benefit from access to visualisations of their own approach to learning?

2.Do you think that using the narrative analysis techniques explored in this project would help learners develop meta-cognition of their own learning style?

3.Do you think that developing meta cognition with regard to their individual learning style would help learners to approach learning differently or with greater insight?

4.As a course designer, would viewing the analysis of many learner log files help you to develop meta-cognition regarding course structures and the validity/accuracy of the adaptivity used to generate course narratives?

5.As a course designer, would filtering the set of learners being analysed based on learning style assist you to detect trends in learner behaviour and thus to modify course elements or structure?

6.As a course designer, would filtering the set of learners based on a classification of log narrative styles assist you to detect trends in learner behaviour and thus to modify course elements or structure?

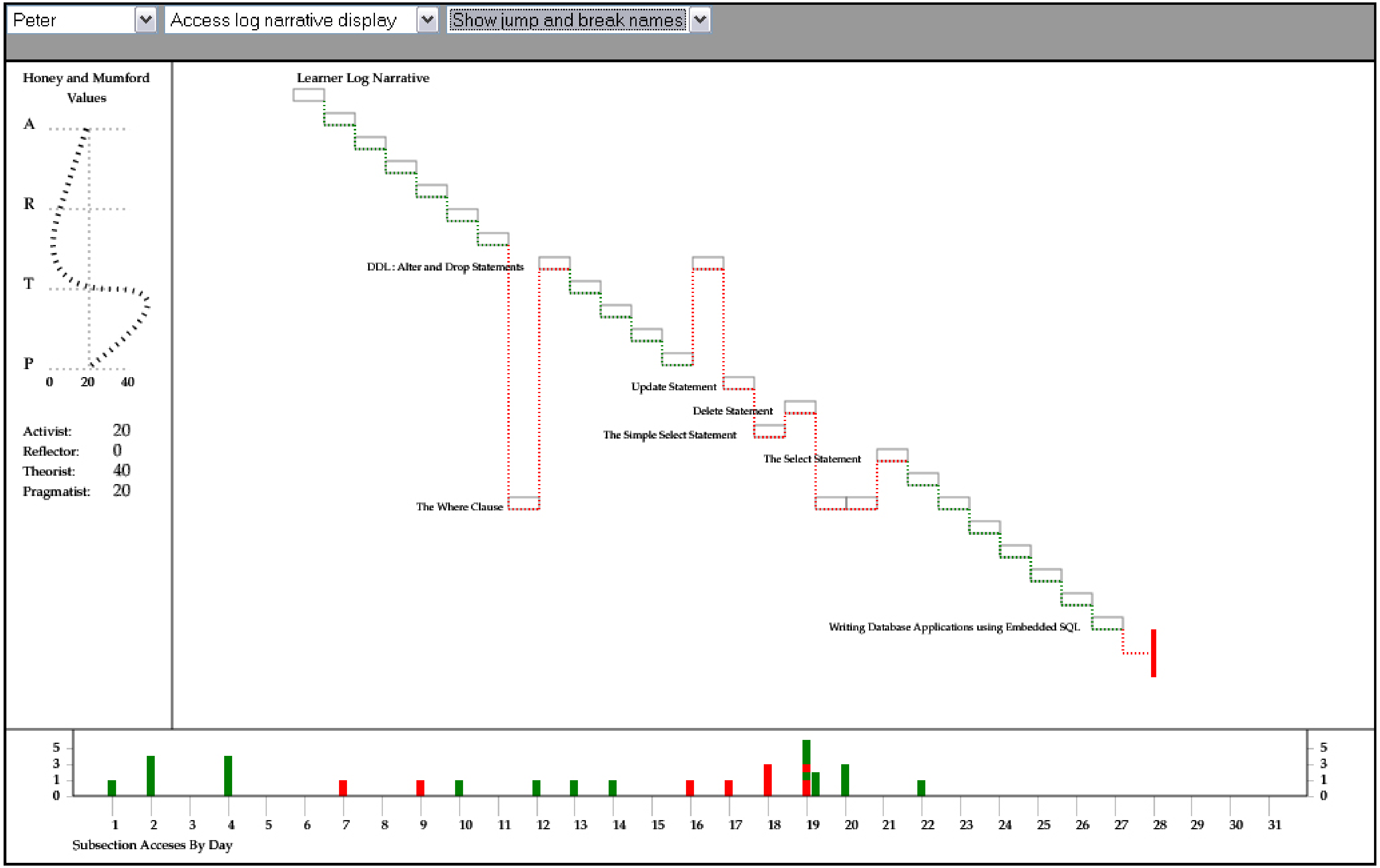
7.As a course designer, would additional functionality, such as enabling changes to content dependencies to be implemented directly from the visualisation, enhance the utility of the narrative analyser significantly?

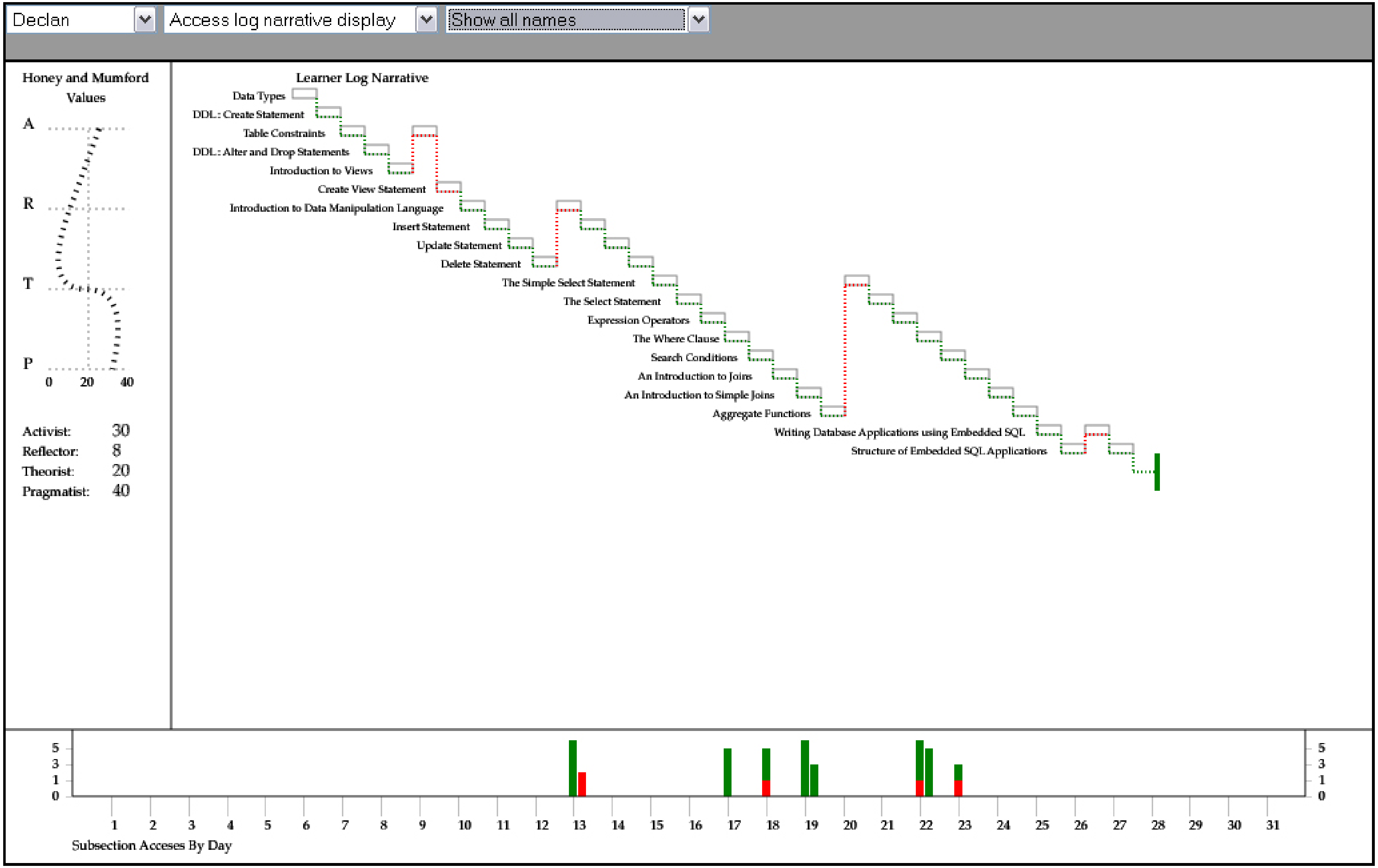
8.Would such a visual 'narrative editor' be of use?

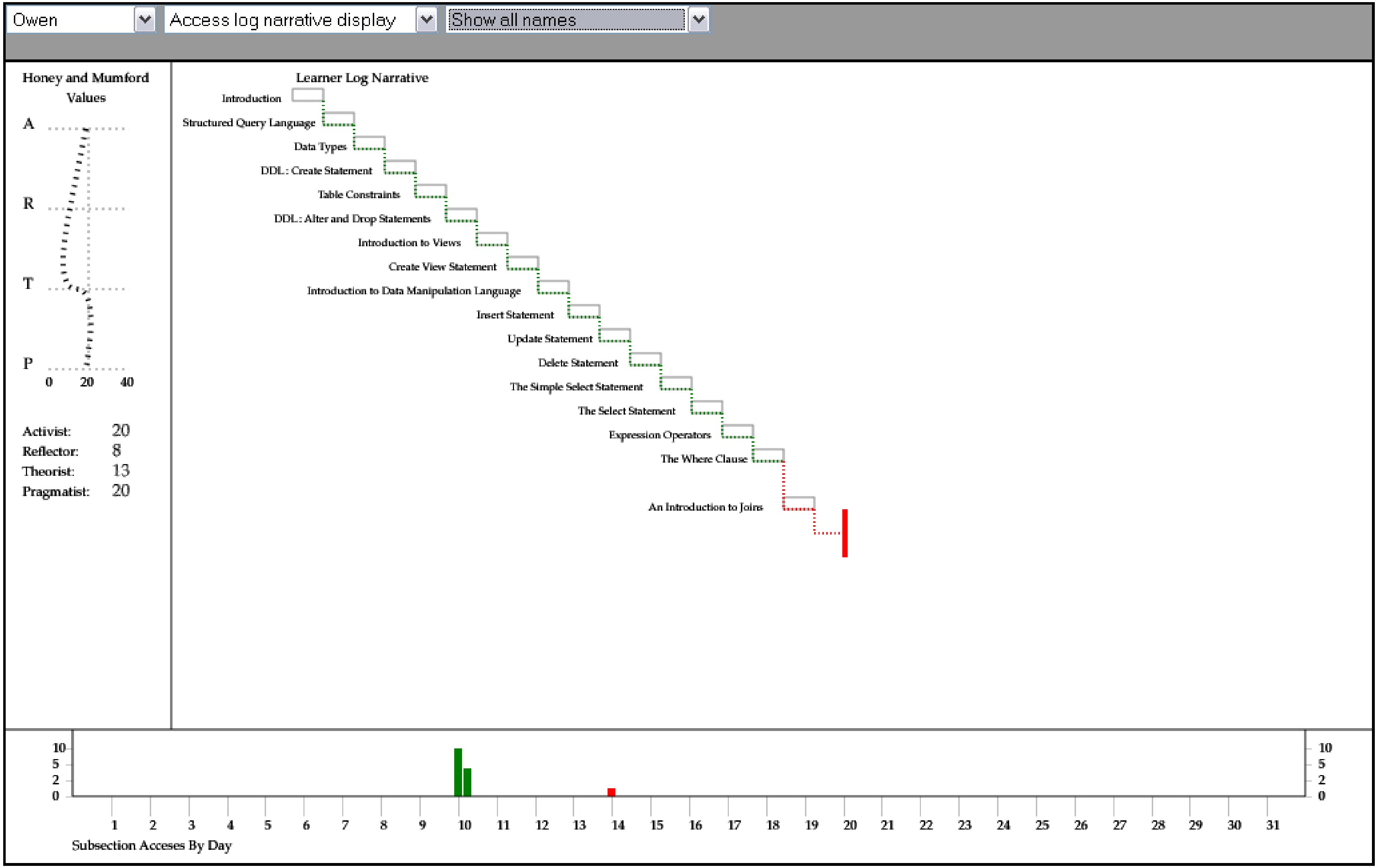
9.As a course designer, would dynamic modification of learning styles either based on post-hoc analysis or real time analysis enhance the personalisation features of the e-Learning system?

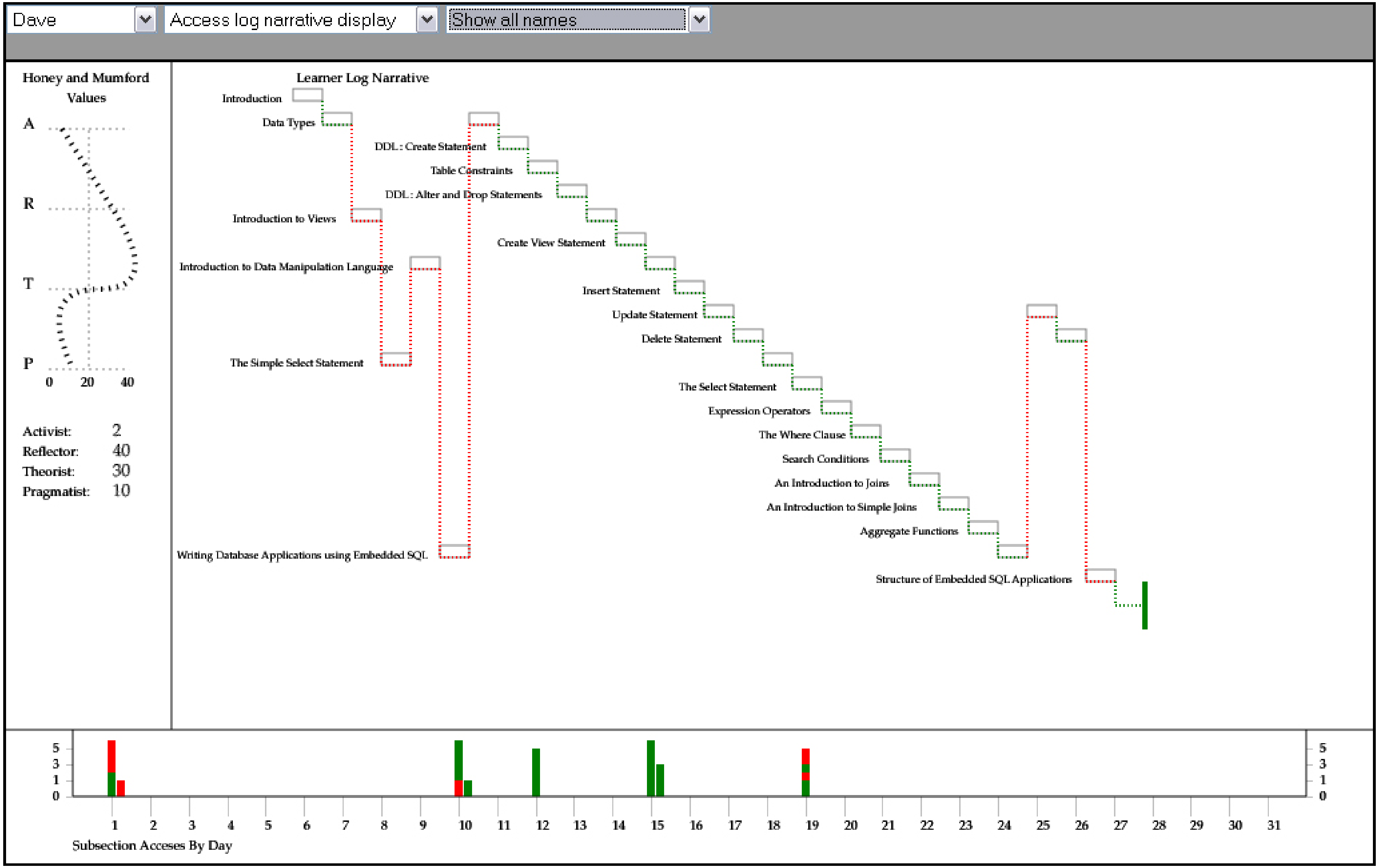
10.Do you think this approach to visualising narrative structures is valid?

Appendix 2. Learner correlated narrative displays.




Declan
Access log narrative display
Show all names
Honey and Mumford Values
A
R
T
P
0
20
40
Activist: 30
Reflector: 8
Theorist: 20
Pragmatist: 40
Learner Log Narrative
Data Types
DDL: Create Statement
Table Constraints
DDL: Alter and Drop Statements
Introduction to Views
Create View Statement
Introduction to Data Manipulation Language
Insert Statement
Update Statement
Delete Statement
The Simple Select Statement
The Select Statement
Expression Operators
The Where Clause
Search Conditions
An Introduction to Joins
An Introduction to Simple Joins
Aggregate Functions
Writing Database Applications using Embedded SQL
Structure of Embedded SQL Applications
5
3
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Subsection Acceses By Day


Owen
Access log narrative display
Show all names
Honey and Mumford Values
A
R
T
P
0
20
40
Activist: 20
Reflector: 8
Theorist: 13
Pragmatist: 20
Learner Log Narrative
Introduction
Structured Query Language
Data Types
DDL: Create Statement
Table Constraints
DDL: Alter and Drop Statements
Introduction to Views
Create View Statement
Introduction to Data Manipulation Language
Insert Statement
Update Statement
Delete Statement
The Simple Select Statement
The Select Statement
Expression Operators
The Where Clause
An Introduction to Joins
10
5
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Subsection Acceses By Day



Dave
Access log narrative display
Show all names
Honey and Mumford Values
A
R
T
P
0
20
40
Activist: 2
Reflector: 40
Theorist: 30
Pragmatist: 10
Learner Log Narrative
Introduction
Data Types
DDL: Create Statement
Table Constraints
DDL: Alter and Drop Statements
Introduction to Views
Create View Statement
Introduction to Data Manipulation Language
Insert Statement
Update Statement
Delete Statement
The Simple Select Statement
The Select Statement
Expression Operators
The Where Clause
Search Conditions
An Introduction to Joins
An Introduction to Simple Joins
Aggregate Functions
Writing Database Applications using Embedded SQL
Structure of Embedded SQL Applications
5
3
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Subsection Acceses By Day

Appendix 3. Core functionality code listings.

makeLogFileEvents()

```
function makeLogFileEvents(learnerName, logFile, courseNarrativeFile, nameDisplayType)
{
        numberOfLogNarrativeJumpEvents = getNumberOfLogNarrativeJumpEvents(logFile, courseNarrativeFile, learnerName);
        var displayHeight = initialDisplayHeight;
        var displayWidth = initialDisplayWidth;
        var blockStartPointX = initialBlockStartPointX;
        var blockStartPointY = initialBlockStartPointY;

        // Scale the display
        var learnerEvents = logFile.getElementsByTagName(learnerName);
        var numberOfLogFileEvents = learnerEvents.length;
        var narrativeSections = courseNarrativeFile.getElementsByTagName("section");
        var numberOfSections = narrativeSections.length;
        var narrativeSubsections = courseNarrativeFile.getElementsByTagName("subsection");
        var numberOfSubsections = narrativeSubsections.length;
        if (numberOfLogNarrativeJumpEvents < 0) numberOfLogNarrativeJumpEvents = (numberOfSubsections - numberOfLogFileEvents);
        var blockLength = Math.round(displayWidth / (numberOfSubsections + numberOfLogNarrativeJumpEvents));
        var blockHeight = Math.round(displayHeight / ((numberOfSubsections + numberOfLogNarrativeJumpEvents) * 2));
        var initialYLineLength = blockHeight;
        var initialXLineLength = blockLength;
        var greenLineStartX = (blockStartPointX + blockLength);
```

```
var greenLineStartY = (blockStartPointY + blockHeight);
var lineStartX = (blockStartPointX + blockLength);
var lineStartY = (blockStartPointY + blockHeight);
var xLineLength = initialXLineLength;

// Set up variables
var elementCount = 0;
var lastSectionName = null;
var logFileSubsectionName = null;
var nextLogFileSectionName = null;
var nextLogFileSubsectionName = null;
var currentLogFileBlockOffset = 0;
var narrativeJumpType = 0;
var monthsToDisplay = 1;
var brokenNarrative = false;
var completeNarrative = false;
accessLog = true;// Used for name display in access log mode

// Start by displaying the heading
elementCount += displayNarrativeStyle("LogFile Events", "Learner Log Narrative", blockStartPointX, blockStartPointY);

// Add markings to the timeline
elementCount += makeTimeline(logFile, learnerName, numberOfLogFileEvents, monthsToDisplay);

// then do the tricky bit
for (logFileEventIndex = 0; logFileEventIndex < numberOfLogFileEvents; logFileEventIndex++)
{
```

```
// Get the section and subSection names
logFileSectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex].childNodes(1).firstChild.nodeValue;
logFileSubsectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex].childNodes(2).firstChild.nodeValue;
lastNarrativeSubsectionName = courseNarrativeFile.getElementsByTagName("subsection")[numberOfSubsections-1].childNodes(0).firstChild.nodeValue;

// Detect a broken or a complete narrative
if (logFileEventIndex == (numberOfLogFileEvents -1))
{
	if (logFileSubsectionName != lastNarrativeSubsectionName) brokenNarrative = true;
	if (logFileSubsectionName == lastNarrativeSubsectionName) completeNarrative = true;
}

if (logFileEventIndex == 0)
{
	previousLogFileSectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex].childNodes(1).firstChild.nodeValue;
	previousLogFileSubsectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex].childNodes(2).firstChild.nodeValue;
}
else
{
	previousLogFileSectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex -1].childNodes(1).firstChild.nodeValue;
	previousLogFileSubsectionName = logFile.getElementsByTagName(learnerName)[logFileEventIndex -1].childNodes(2).firstChild.nodeValue;
}

// Work out the narrative offsets of the logFile events
var logFileSubsectionOffset = getOffset(courseNarrativeFile, logFileSubsectionName, logFileSectionName, numberOfSubsections);
var previousLogFileSubsectionOffset = getOffset(courseNarrativeFile, previousLogFileSubsectionName, previousLogFileSectionName, numberOfSubsections);
var jumpLength = (logFileSubsectionOffset - previousLogFileSubsectionOffset);
```

```
var offsetY = (blockHeight * 2);

// Set the start points
if ((jumpLength == 0) && (logFileSubsectionName == previousLogFileSubsectionName) && (logFileEventIndex != 0))
{
	blockStartPointY = (blockStartPointY - (blockHeight * 2));
}
if (jumpLength > 1)// Forward jump
{
	for (i=0; i < (jumpLength  -1); i++) offsetY += (blockHeight * 2);
	blockStartPointY = ((blockStartPointY + offsetY) - (blockHeight * 2));
	narrativeJumpType = 1;
}
if (jumpLength < 0)// Backwards jump
{
	for (i=0; i >= jumpLength; i--) offsetY -= (blockHeight * 2);
	blockStartPointY = ((blockStartPointY + offsetY) - (blockHeight * 2));
	narrativeJumpType = 1;
}

yLineLength = offsetY;

// Display a block
elementCount += displayBlock("LogFile Events", blockStartPointX, blockStartPointY, blockLength, blockHeight);

// Handle the display of lines
if ((jumpLength > 1) || (jumpLength < 0))
```

```
{
	elementCount += displayLine("LogFile Events", "Y", "red", lineStartX, lineStartY, yLineLength, xLineLength, numberOfSubsections);
	elementCount += displayLine("LogFile Events", "X", "red", lineStartX, (lineStartY + yLineLength), yLineLength, xLineLength, numberOfSubsections);
}
else
{
	if ((jumpLength == 0) && (logFileEventIndex != 0))
	{
		elementCount += displayLine("LogFile Events", "X", "red", lineStartX, lineStartY, yLineLength, xLineLength, numberOfSubsections);
	}
	else // default action
	{
		elementCount += displayLine("LogFile Events", "Y", "green", lineStartX, lineStartY, yLineLength, xLineLength, numberOfSubsections);
		elementCount += displayLine("LogFile Events", "X", "green", lineStartX, (lineStartY + yLineLength), yLineLength, xLineLength,
												numberOfSubsections);
	}
}

if (brokenNarrative == true)
{
	elementCount += displayLine("LogFile Events", "Y", "red", (lineStartX + xLineLength), (lineStartY + yLineLength), (initialYLineLength *2),
											xLineLength, numberOfSubsections);
	elementCount += displayLine("LogFile Events", "X", "red", (lineStartX + xLineLength), (lineStartY + (yLineLength + (initialYLineLength * 2))),
										initialYLineLength, xLineLength, numberOfSubsections);
	var brokenNarrativeMarker = "<line style=\' stroke:red; stroke-width:4\' x1=\'" +(lineStartX + (xLineLength *2))+"\' y1=\'" +(lineStartY +
		(yLineLength))+"\' x2=\'" +(lineStartX + (xLineLength *2))+"\' y2=\'"+(lineStartY + (yLineLength + (initialYLineLength *4)))+"\'/>";
	AddSVG(brokenNarrativeMarker, "LogFile Events");
	elementCount++;
}
```

```
if (completeNarrative == true)
{
    elementCount += displayLine("LogFile Events", "Y", "green", (lineStartX + xLineLength), (lineStartY + yLineLength), (initialYLineLength *2),
                                                                                        xLineLength, numberOfSubsections);
    elementCount += displayLine("LogFile Events", "X", "green", (lineStartX + xLineLength), (lineStartY + (yLineLength + (initialYLineLength * 2))),
                                                                          initialYLineLength, xLineLength, numberOfSubsections);
    var brokenNarrativeMarker = "<line style=\' stroke:green; stroke-width:4\' x1=\'" +(lineStartX + (xLineLength *2))+"\' y1=\'" +(lineStartY +
                (yLineLength))+"\' x2=\'" +(lineStartX + (xLineLength *2))+"\' y2=\'"+(lineStartY + (yLineLength + (initialYLineLength *4)))+"\'/>";
    AddSVG(brokenNarrativeMarker, "LogFile Events");
    elementCount++;
}

// Display subSectionNames if selected
if (nameDisplayed(logFileSubsectionName) == false)
{
    if (nameDisplayType == 1)
    elementCount += displaySubsectionName("LogFile Events", accessLog, logFileSubsectionName, (blockStartPointX + blockLength), (blockStartPointY +
                                                                                        blockHeight), blockStartPointX);
    if ((nameDisplayType == 2) && ((jumpLength > 1) || (jumpLength < 0) || (brokenNarrative == true)))
    elementCount += displaySubsectionName("LogFile Events", accessLog, logFileSubsectionName, (blockStartPointX + blockLength), (blockStartPointY +
                                                                                        blockHeight), blockStartPointX);
}

// Add an event to the timeline
elementCount += makeTimelineEvent(learnerName, logFile, monthsToDisplay, narrativeJumpType, numberOfLogFileEvents, logFileEventIndex);

// Update variables for next loop
greenLineStartX = greenLineStartX + blockLength;
```

```
                greenLineStartY = greenLineStartY + (blockHeight + yLineLength);
                currentLogFileBlockOffset = 0;
                ylineLength = initialYLineLength;
                xLineLength = initialXLineLength;
                lineStartX = (blockStartPointX + blockLength);
                lineStartY = (blockStartPointY + blockHeight);
                blockStartPointX = (blockStartPointX + blockLength);
                blockStartPointY = (blockStartPointY + (blockHeight * 2));
                narrativeJumpType = 0;
        } // End for(logFileEventIndex)

        accessLog = 0; // Reset this
        if (elementCount == 1) elementCount++;
        return (elementCount-1);
}// End makeLogFileEvents
```

makeNarrativeEvents()

```
function makeNarrativeEvents(courseNarrativeFile)
{
        var displayHeight = initialDisplayHeight;
        var displayWidth = initialDisplayWidth;
        var blockStartPointX = initialBlockStartPointX;
        var blockStartPointY = initialBlockStartPointY;
        var narrativeSections = courseNarrativeFile.getElementsByTagName("section");
```

```
var numberOfSections = narrativeSections.length;
var narrativeSubsections = courseNarrativeFile.getElementsByTagName("subsection");
var numberOfSubsections = narrativeSubsections.length;
var xLineLength = Math.round(displayWidth/(numberOfSubsections * 8));
var yLineLength = Math.round(displayHeight/(numberOfSubsections * 2));
var blockLength = Math.round(displayWidth/(numberOfSubsections + numberOfLogNarrativeJumpEvents));
var blockHeight = Math.round((displayHeight/numberOfSubsections) - yLineLength);
var greenLineStartX = (blockStartPointX + blockLength);
var greenLineStartY = (blockStartPointY + blockHeight);
var greenLineCount = 1;
var lastSectionName = null;
var currentsectionName = null;
var elementCount = 0;

// Display title
elementCount += displayNarrativeStyle("Course Narrative Events", "SQL Course Narrative", blockStartPointX, blockStartPointY);

for (narrativeFileIndex = 0; narrativeFileIndex < numberOfSubsections; narrativeFileIndex++)
{
	var narrativeSubsectionName = courseNarrativeFile.getElementsByTagName("subsection")[narrativeFileIndex].childNodes(0).firstChild.nodeValue;

	// Build display
	currentSectionName = lastSectionName;
	lastSectionName = displaySectionName("Course Narrative Events", narrativeFileIndex, lastSectionName, blockStartPointX, greenLineStartY,courseNarrativeFile);
	if (lastSectionName != currentSectionName) elementCount++;// only update if new name has been displayed
	if (narrativeFileIndex <= (numberOfSubsections -2))//avoid drawing an extra line at the bottom of the list
	{
```

```
                elementCount += displayLine("Course Narrative Events", "Y", "green", greenLineStartX, greenLineStartY, yLineLength, xLineLength,
                                                                                                        numberOfSubsections);
            }
            elementCount += displayBlock("Course Narrative Events", blockStartPointX, blockStartPointY, blockLength, blockHeight);
            elementCount += displaySubsectionName("Course Narrative Events", accessLog, narrativeSubsectionName, greenLineStartX, greenLineStartY,
                                                                                                        blockStartPointX);

            // Update variables for next loop
            blockStartPointX = blockStartPointX + blockLength;
            blockStartPointY = blockStartPointY + (blockHeight + yLineLength);
            greenLineStartX = greenLineStartX + blockLength;
            greenLineStartY = greenLineStartY + (blockHeight + yLineLength);
        }
        return (elementCount-1);//elements added
}// End makeNarrativeEvents
```

makeMentebarCurve()

```
function makeMentebarCurve(learnerModelDoc)
{
        var xOffset         =  40;
        var activistX       =  xOffset + parseInt(learnerModelDoc.getElementsByTagName("activist")[0].firstChild.nodeValue)
        var reflectorX      =  xOffset + parseInt(learnerModelDoc.getElementsByTagName("reflector")[0].firstChild.nodeValue)
        var theoristX       =  xOffset + parseInt(learnerModelDoc.getElementsByTagName("theorist")[0].firstChild.nodeValue)
        var pragmatistX  =  xOffset + parseInt(learnerModelDoc.getElementsByTagName("pragmatist")[0].firstChild.nodeValue)

        var mentebarCurve = "<path  fill=\'none\' stroke=\'#000000\' stroke-width=\'4\' stroke-miterlimit=\'2\' d= \'M "+ activistX +" 50 T" + reflectorX +" 110  T "+ theoristX +"
```

```
170 T "+ pragmatistX +" 230\'>"
	AddSVG(mentebarCurve, "Mentebar Curve");
	displayText ="<text x=\'80\' y=\'280\' style=\'font-family:palatino; font-size:13\'>"+(activistX - xOffset)+" </text>"
	AddSVG(displayText, "Mentebar Curve");
	displayText ="<text x=\'80\' y=\'295\' style=\'font-family:palatino; font-size:13\'>"+(reflectorX - xOffset)+" </text>"
	AddSVG(displayText, "Mentebar Curve");
	displayText ="<text x=\'80\' y=\'310\' style=\'font-family:palatino; font-size:13\'>"+(theoristX - xOffset)+" </text>"
	AddSVG(displayText, "Mentebar Curve");
	displayText ="<text x=\'80\' y=\'325\' style=\'font-family:palatino; font-size:13\'>"+(pragmatistX - xOffset)+" </text>"
	AddSVG(displayText, "Mentebar Curve");

	return (4);// The number of events to be removed
}
```

makeTimeLine()

```
function makeTimeline(logFile, learner, numberOfLogFileEvents, monthsToDisplay)
{
	var divisionsToAdd = (32 * monthsToDisplay);
	var eachDivision = (924/divisionsToAdd); //the length of each division of the timeline in pixels
	var displayPoint = (50);
	var maxAccessesByDay = 0;
	var accessesThisDay = 0;

	var legend = "<text x=\'50\' y=\'590\' style=\'font-family:palatino; font-size:10\'>Subsection Acceses By Day </text>";
	AddSVG(legend, "LogFile Events");
```

```
// Set the vertical scaling
for (logFileEventIndex=0; logFileEventIndex < numberOfLogFileEvents; logFileEventIndex++)
{
	var thisDay = logFile.getElementsByTagName(learner)[logFileEventIndex].childNodes(3).firstChild.nodeValue;
	if (logFileEventIndex == 0) var previousDay = 32; // no match in the month
	else var previousDay = logFile.getElementsByTagName(learner)[logFileEventIndex -1].childNodes(3).firstChild.nodeValue;
	if (previousDay == thisDay) accessesThisDay ++;
	else
	{
		accessesThisDay = 1;
	}
	if (maxAccessesByDay < accessesThisDay) maxAccessesByDay = accessesThisDay;
}

if (maxAccessesByDay > 15)
{
	markHeight = 6;
	columnOneValue = 10;
	columnTwoValue = 20;
	var heightMarkerValueZero = "<text x=\'35\' y=\'553\' style=\'font-family:palatino; font-size:10\'>0</text>"
	AddSVG(heightMarkerValueZero, "LogFile Events");
	var heightMarkerValueOne = "<text x=\'35\' y=\'541\' style=\'font-family:palatino; font-size:10\'>2</text>"
	AddSVG(heightMarkerValueOne, "LogFile Events");
	var heightMarkerValueTwo = "<text x=\'35\' y=\'529\' style=\'font-family:palatino; font-size:10\'>5</text>"
	AddSVG(heightMarkerValueTwo, "LogFile Events");
	var heightMarkerValueThree = "<text x=\'35\' y=\'517\' style=\'font-family:palatino; font-size:10\'>10</text>"
```

```
        AddSVG(heightMarkerValueThree, "LogFile Events");
        var heightMarkerValueZero = "<text x=\'984\' y=\'553\' style=\'font-family:palatino; font-size:10\'>0</text>"
        AddSVG(heightMarkerValueZero, "LogFile Events");
        var heightMarkerValueOne = "<text x=\'984\' y=\'541\' style=\'font-family:palatino; font-size:10\'>2</text>"
        AddSVG(heightMarkerValueOne, "LogFile Events");
        var heightMarkerValueTwo = "<text x=\'984\' y=\'529\' style=\'font-family:palatino; font-size:10\'>5</text>"
        AddSVG(heightMarkerValueTwo, "LogFile Events");
        var heightMarkerValueThree = "<text x=\'984\' y=\'517\' style=\'font-family:palatino; font-size:10\'>10</text>"
        AddSVG(heightMarkerValueThree, "LogFile Events");
}
else
{
        markHeight = 12;
        columnOneValue = 5;
        columnTwoValue = 10;
        var heightMarkerValueZero = "<text x=\'35\' y=\'553\' style=\'font-family:palatino; font-size:10\'>0</text>"
        AddSVG(heightMarkerValueZero, "LogFile Events");
        var heightMarkerValueOne = "<text x=\'35\' y=\'541\' style=\'font-family:palatino; font-size:10\'>1 </text>"
        AddSVG(heightMarkerValueOne, "LogFile Events");
        var heightMarkerValueTwo = "<text x=\'35\' y=\'529' style=\'font-family:palatino; font-size:10\'>3 </text>"
        AddSVG(heightMarkerValueTwo, "LogFile Events");
        var heightMarkerValueThree = "<text x=\'35\' y=\'517\' style=\'font-family:palatino; font-size:10\'>5 </text>"
        AddSVG(heightMarkerValueThree, "LogFile Events");
        var heightMarkerValueZero = "<text x=\'984\' y=\'553\' style=\'font-family:palatino; font-size:10\'>0</text>"
        AddSVG(heightMarkerValueZero, "LogFile Events");
        var heightMarkerValueOne = "<text x=\'984\' y=\'541\' style=\'font-family:palatino; font-size:10\'>1 </text>"
        AddSVG(heightMarkerValueOne, "LogFile Events");
```

```
		var heightMarkerValueTwo = "<text x=\'984\' y=\'529' style=\'font-family:palatino; font-size:10\'>3 </text>"
		AddSVG(heightMarkerValueTwo, "LogFile Events");
		var heightMarkerValueThree = "<text x=\'984\' y=\'517\' style=\'font-family:palatino; font-size:10\'>5 </text>"
		AddSVG(heightMarkerValueThree, "LogFile Events");
}

// add the markings and day of month
for (i=0; i <= (divisionsToAdd -2); i++) //timeline markings
{
		var thisDivision = Math.round(displayPoint + eachDivision);
		var markerLine = "<line x1=\'" + thisDivision +"\' y1=\'550\' x2=\'" + thisDivision +"\' y2=\'565\' />"
		AddSVG(markerLine, "LogFile Events");
		var day = "<text x=\'"+ thisDivision +"\' y=\'575\' style=\'font-family:palatino; font-size:10\'>"+(i + 1)+" </text>"
		AddSVG(day, "LogFile Events");
		displayPoint = displayPoint + eachDivision;
}

var scaleLine = "<line x1=\'45\' y1=\'550\' x2=\'50\' y2=\'550\'/>"
AddSVG(scaleLine, "LogFile Events");
var scaleLine = "<line x1=\'45\' y1=\'538\' x2=\'50\' y2=\'538\'/>"
AddSVG(scaleLine, "LogFile Events");
var scaleLine = "<line x1=\'45\' y1=\'526\' x2=\'50\' y2=\'526\'/>"
AddSVG(scaleLine, "LogFile Events");
var scaleLine = "<line x1=\'45\' y1=\'514\' x2=\'50\' y2=\'514\'/>"
AddSVG(scaleLine, "LogFile Events");
var scaleLine = "<line x1=\'974\' y1=\'550\' x2=\'979\' y2=\'550\'/>"
AddSVG(scaleLine, "LogFile Events");
```

```
	var scaleLine = "<line x1=\'974\' y1=\'538\' x2=\'979\' y2=\'538\'/>"
	AddSVG(scaleLine, "LogFile Events");
	var scaleLine = "<line x1=\'974\' y1=\'526\' x2=\'979\' y2=\'526\'/>"
	AddSVG(scaleLine, "LogFile Events");
	var scaleLine = "<line x1=\'974\' y1=\'514\' x2=\'979\' y2=\'514\'/>"
	AddSVG(scaleLine, "LogFile Events");

	return (((divisionsToAdd * 2)-2) +17) // Add up all the elements added
}// end makeTimeLine()
```

makeTimeLineEvent()

```
function makeTimeLineEvent(learner, logFile, monthsToDisplay, narrativeJumpType, numberOfLogFileEvents, logFileEventIndex)
{
	var timeScale = null;
	var strokeWidth = 6;
	var startX = 0;
	var startY = 550;

	if (monthsToDisplay == 1)
	{
		var eachDivision = (924/32); //the length of each time division in pixels
		var lastMonth = logFile.getElementsByTagName(learner)[numberOfLogFileEvents -1].childNodes(2).firstChild.nodeValue;
		var thisAccessDay = logFile.getElementsByTagName(learner)[logFileEventIndex].childNodes(3).firstChild.nodeValue;
```

```
// Work out the markHeight and key points
startX = Math.round(50 + (eachDivision * thisAccessDay));

if (logFileEventIndex == 0) var lastAccessDay = 32; //so that it can not match any day of the month
else var lastAccessDay = logFile.getElementsByTagName(learner)[logFileEventIndex -1].childNodes(3).firstChild.nodeValue;

if (lastAccessDay == thisAccessDay) eventsPerDay ++; else eventsPerDay = 1;

if (eventsPerDay > 1)
{
	if (eventsPerDay > columnOneValue)
	{
		if (eventsPerDay > columnTwoValue)
		{
			startX = (startX + ((strokeWidth *2) +1));
			startY = (550 - ((markHeight/2) * (eventsPerDay -(columnTwoValue +1))));
		}
		else
		{
			startX = (startX + (strokeWidth +1));
			startY = (550 - ((markHeight/2) * (eventsPerDay - (columnOneValue +1))));
		}
	}
	else startY = (550 - ((markHeight/2) * eventsPerDay));
}
else startY = 550; // First event today
```

```
		var endY = startY - markHeight;

		// Assign colour to event
		switch (narrativeJumpType)
		{
			case 0: var markColour = "green"; break;
			case 1: var markColour = "red"; break;
			case 2: var markColour = "blue"; break;
		}

		var mark= "<line style=\'stroke:"+markColour+"; stroke-width:"+strokeWidth+"\'  x1=\'"+startX+"\' y1=\'"+startY+"\' x2=\'"+startX+"\' y2=\'"+endY+"\'/>";
		AddSVG(mark, 'LogFile Events');
	}
	return (1); // Number of events added
}// end makeTimeLineEvent()
```

Appendix 4. Evaluation responses.

Questions	Interview One	Interview Two	Interview Three
Did learner "Dave" complete the suggested course?	He did – by looking at the narrative log	He did. It shows you the power of visual data mining – we would not have been able to determine that simple fact easily without it.	He did. The green icon clearly indicates success.
Describe learner "Dave"'s learning style with reference to his log narrative and Honey and Mumford classification?	He assessed Dave's learning style by reading the discrete value figures rather than looking at the curve and was surprised when this was pointed out. The learning curve.	Noses – a full body nose emerged as an interpretation of the H & M curve " a full bodied Roman nose"	He likes to check the concepts first, then go back and work through the detail. He is dominantly a theorist.
Does learner "Declan"'s access pattern suggest a strong Activist/Pragmatist to you?	Yes, but it was kind of hard to see. Maybe it would take time to become proficient at recognising different learning styles in the access pattern, if indeed the two correlate. If they don't that is interesting in itself...	Is there a more fundamental question here – teachers like to focus on content not process. Do we just provide the evidence or do we provide interpretation?	This raised the issue of redefinition of the learner style and the content dependant nature of some narrative analysis.
What does user "Peter"'s incomplete narrative suggest to you?	He had learnt enough. He had more time to finish but decided not to complete the last subsection.	He did not need to know this bit. He knows what questions are on the exam?	He went far enough. He was only one step from completion of the course narrative so he must have made a decision to stop. Having done the rest of the course there should have been no reason not to complete the last subsection.
Which learner employs revision extensively as part of their learning style?	I suppose revision would equate to backward jumps.. Declan seems to be the most revision oriented. Stalagmites and stalactites shape was suggested by narrative jumps.	Declan would be closest. Dave plotted a course by exploring but then went back and followed the course sequentially.	Declan. Dave seemed to at first glance but when you look again he did the rest of the course sequentially with very little revision.

What might user “Owen”'s log narrative suggest, if anything?	Either he knows the subjects that follow or he decided he did not need to know them.	He is well rounded in H&M terms. Last access was a month before the end. Maybe he started becoming more pragmatic and found other learning sources.	Perhaps he got run over by a bus? This prompted a discussion about posing questions nobody asked before.
What does user “Declan”'s log narrative time line tell us about his learning style as contrasted with user “Peter”'s log narrative time line.	Peter is more spread out but the pattern seems to be alike. Both are very different to Owens for instance.	The pattern is similar even though the timescale the accesses are spread out over is longer.	Both seem to approach the process of learning in a similar manner but with a difference in time priority given to the task perhaps.
What would you change about the visualisation in order to improve it's utility?	What if the course narrative was visible in some way at all times, maybe ghosted out... Selecting multiple users based on drawing a curve in the H & M window?	The main issue with referencing the course narrative is with narrative termination. If the narrative continued after the terminator greyed out or something. Fish-eye view to magnify events? Display of multiple users simultaneously. Show me “all pragmatist users”. Averaged results display.	I would like to see some mechanism or technique which allowed me to correlate many users or many narratives. How you might do this I'm not sure! Some way to integrate and display changes to a learner's learning style definition would be useful. Being able to tell how long a user was actually looking at the screen would be interesting to extend the analysis in a qualitative way.
What would you change about the visualisation in order to improve clarity?	If anything you should be careful not to lose the clarity it has now by adding too many new types of information. Each subset of the possible displays should be clear in it's own right	Maybe to add some more colour coding to the narrative model, for instance when averaging results for a selection of learners.	Keep it the way it is, the document look works well.

Do you think learners could benefit from access to visualisations of their own approach to learning?	Insight is never wasted. We inform our future decisions based on insight into our past successes and failures. In this case it might be that a learner would change the way he tackled the next course based on insight gained from the analysis of this one, or modify his learning style to better reflect his preferences during this one.	If learners know that their access pattern is going to be analysed would that change the way they access?	Yes. We do not often look at our behaviour from outside.
Do you think that using the narrative analysis techniques explored in this project would help learners develop meta-cognition of their own learning style?	Developing meta cognition could help designers by enabling learners to give more insightful feedback about the course.	Definite possibilities. You need to be careful that this technology does not create "expert learners" "The whole world is not going to become more theoretical just because you are."	Perhaps, if they have the interest to look at the analysis, they probably have some cognitive insight anyway. Some learners might access it simply because it is there and gain insights almost by accident, but that would be positive too...
Do you think that developing meta cognition with regard to their individual learning style would help learners to approach learning differently or with greater insight?	Yes so long as it is not seen as a distraction from the main learning goal.	As above, the expert learner is an issue.	Yes if it is applied to giving the system a more accurate model. The cultural factors which can distort the definition – what do I want to be like rather than what am I like? A business background might give rise to a lot of pragmatic activists!
As a course designer, would viewing the analysis of many learner log files help you to develop meta-cognition regarding course structures and the validity/accuracy of the adaptivity used to generate course narratives?	Course designers are in the dark about how learning styles are actually related to the real world usage of the system. Again feedback about this would be very useful. Access logs are generated, filed, and archived... We need a tool like this to make sense of them.	The term meta cognition does not fit here. However the idea of trying to validate the output of the adaptive system by reference to the user's experience of that narrative is an exciting idea...	A tool like this would be very useful although it might take some time to be able to get the most out of it. Maybe it should incorporate some knowledge of the high level correlations we have been talking about.

As a course designer, would filtering the set of learners being analysed based on learning style assist you to detect trends in learner behaviour and thus to modify course elements or structure?	Certainly with regard to modifying courses for future use. The narrative analysis does seem to have the potential to highlight particular subsections that might need further work. Establishing the correlation between learning styles and real behaviour would be very interesting...	Its very useful. Some form of scripting language in order to define selection groups or view filters. Or select by circling two subsections and look at all learners for those subsections.	Indeed, a way of testing the match between learning styles and actual use would be interesting, and might highlight new issues with the behaviour of learners. For instance, to look at usage patterns at 'pressure points' in the calender – like just before exams...
As a course designer, would filtering the set of learners based on a classification of log narrative styles assist you to detect trends in learner behaviour and thus to modify course elements or structure?	If it can be done – yes.	How many people jumped back from section X? Or how many people broke their narrative here? These could be filters of the view. Or, which subsections featured revision jumps most often might indicate difficulties encountered on a broad basis.	If we could begin to see how new concepts are dealt with then it would be interesting to be able to resequence key concepts and validate them
As a course designer, would additional functionality, such as enabling changes to content dependencies to be implemented directly from the visualisation, enhance the the utility of the narrative analyser significantly?	It could be very difficult to integrate this functionality so it may be better to see the narrative analyser as a tool to be used in conjunction with the course composition tools.	As part of the course development yes, if there is a sufficient data then it could be used to simulate users and exercise the system. The idea of 'conceptual load' was introduced which might be a factor in the diagnosis of problem narrative subsections.	Again validation of narratives using a model of user access behaviour is very interesting. Gaining some quantitative information about the success or failure of an individual narrative against the learner history could yield interesting statistics. The idea of a larger data model being kept for a learner.
Would such a visual 'narrative editor' be of use?	Yes, it could give another viewpoint for course designers to consider –	Yes.	Yes but you ultimately need a narrative analyser and a narrative validation tool which could be used at design time.

As a course designer, would dynamic modification of learning styles either based on post-hoc analysis or real time analysis enhance the personalisation features of the e-Learning system?	This kind of analysis would at least be based on what people actually did rather than what they said they did... Including a visual indication of the changes to the learner model could improve the understanding of the changing priorities of learners over time.	The data-mining problem. This is a very interesting issue but difficult to solve I think.. The learner might be using the redefinition of the learning model as a form of index to select content...thus forming a nasty loop.	Either modification or simply some form of validation would be useful. Learners exercise self-deception when it comes to defining their learning style.
Do you think this approach to visualising narrative structures is valid?	It really sheds a lot of light on the issue. Course designers could benefit enormously. A tool like this really could close the loop. Extra meta data could be added to the content to aid with later analysis.	Yes. There are many possibilities for how it could be developed.	There are so many ways this tool could develop you need to be careful to choose the best direction.