

Ariadne: a Java-based guided tour system for the World Wide Web

[Jesper Jühne](#)^a, [Anders T. Jensen](#)^b and [Kaj Grønbæk](#)^c

^a*Tele Danmark Internet,
Olof Palmes Allé 36, DK-8200 Århus N, Denmark
jesper@jyhne.dk*

^b*LEGO A/S, SPU-Darwin,
Kløvermarken 120, DK-7190 Billund, Denmark
anders@digi.lego.com*

^c*Department of Computer Science, University of Aarhus,
Ny Munkegade, Bldg. 540, DK-8000 Århus C, Denmark
kgronbak@daimi.aau.dk*

Abstract

This paper presents a Guided tour system for the WWW, called Ariadne, which implements the ideas of trails and guided tours, originating from the hypertext field. Ariadne appears as a Java applet to the user and it stores guided tours in a database format separated from the WWW documents included in the tour. The main advantages of the system compared to similar systems are: (1) an independent user interface which does not affect the layout of the documents being part of the tour, (2) branching tours where the user may follow alternative routes, (3) composition of existing tours into aggregate tours, (4) overview map with indication of which parts of a tour have been visited and support for getting back on track. Ariadne is available as a research prototype, and it has been tested among a group of university students as well as casual users on the Internet.

Keywords

Guided tours; Navigation; Hypermedia; Presentation

1. Introduction

This paper presents a Guided Tours System for the WWW [1], called Ariadne. [1] A guided tour is a prepared (possibly branching) sequence of documents, which can be presented to the user by single stepping through a graph. We claim that it is useful to introduce guided tours on the WWW.

The World Wide Web is at present the most widespread electronic medium for the ordinary user. Millions of documents exist on the WWW. This enormous information corpus is one of the great advantages of the WWW. It is possible to obtain information about all kinds of topics. Another advantage of the WWW is the distributed nature of the information; the owner of a document has full control and is the only one with editorial options. These strengths, however, also become a problem to the user. The amount of information and the users' freedom to organise it, as they like, makes the Web overwhelming and at times hard to navigate in.

The owner's control of their documents creates a problem. It is not possible for other users of the Web to reorganise the documents for their own purposes. Even though it is possible to link to another person's document, it is not possible to make a link from another person's document. This makes it difficult to make a fluid presentation based on documents, that you do not own.

We claim that external hypermedia structures, such as independent links, collections, and guided tours would help the users of the WWW overcome many of these problems. Much research and development has already been made to add external structures to the WWW, such as links and collections. Hyper-G/HyperWave [10], DHM/WWW [6], Microcosm DLS [4]. Some of these projects have also added guided tour support to the WWW. Pooh's Guided Tours Service [8], Footsteps [11], and Walden's Paths [5] are all projects, which have added guided tours to the WWW. All of these efforts in providing value-adding services for the WWW aim to make it easier for the prospective users of the WWW to find their way through relevant information.

Ariadne is inspired both from the classical hypertext research [2, 3, 12, 13] and the recent initiatives on the WWW, and it attempts to take further steps in the direction of supporting users in navigating through prepared presentations of subjects.

The Ariadne prototype is available at <http://www.daimi.aau.dk/~jjuhne/Ariadne/> for free trial.

2. The merits of Ariadne

Ariadne has been carefully designed to overcome many of the problems and limitations of existing Guided Tour systems.

In particular, the following four issues have played a main role in the design of Ariadne:

1. Independent user interface — keeps the presentation of the WWW pages as if they were displayed in the browser.
2. Branching tours — allow the user to follow alternative routes in a guided tour.
3. Composition of tours — allows a tour to be an aggregation of other tours.
4. Navigation overview — a graphical presentation of the guided tour and its history.

We will here briefly argue for these choices. We have examined a number of existing guided tour systems: Pooh's Guided Tours Service [8], Walden's Paths [5] and Footsteps [11]. These systems all apply a *pre-processing* approach in order to add guided tour navigation on to the original WWW documents. They compile navigation button code into the HTML specification of the original document. In this way documents appear in the browser with navigation buttons at the top the page or in a separate frame. We think this is a sub optimal solution since these systems alter the layout and presentation of the information that the original author prepared. In contrast, pages in the Ariadne system are displayed in a window by themselves and they present the WWW documents in a separate browser window as they would have been presented if the user had retrieved them independently of Ariadne. We see this as an advantage for the user, because this makes the appearance of a WWW document uniform and independent of the context in which it is displayed. We believe the uniform presentation of WWW documents adds to the user's overview.

The examined guided tour systems on the WWW only support linear tours, i.e. authors of tours cannot create a tour that contains branching routes. If an author of a guided tour wants to create a tour where the readers should be able to choose, e.g. between different subtopics, this would be difficult only having support for linear tours. If the author would like to create two different sub tours in a tour, e.g. one describing a problem in detail and one just giving the reader an overview of a topic, the branching option would be handy. The author using the Ariadne system can create branching routes in an easy way.

The user of a guided tour system should have the ability to reuse tours already created as parts of their own tours. This is supported by the backend of the Ariadne system, but it has not yet been supported in the user interface. When this service becomes available, it will help users in reusing other authors' work.

In order to take full advantage of the external guided tour structure, the user should be able to view the tour while following the tour. Furthermore, the user should be able to view — in an easy way — the nodes in the tour that have been visited and where the user is currently in the tour. Finally, the author of a tour should be able to help the user by adding names or/and annotation to the nodes. This is all supported in the Ariadne system. It helps users keep their orientation in the tour, e.g. if the user has lost orientation because s/he has gone off the tour by following links from a page in the tour, a quick look at the graphical presentation of the tour helps the user regain the focus. The annotations give the author the possibility to add comments to WWW documents in the tour. This could be used in a variety of use situations, e.g. if the author wants to review the sites in the tour, draw the readers attention to a specific topic in the document or — in an educational situation & 51; adding questions to the topics touched in the document.

3. System architecture and user interface

This section describes the architecture and the user interface of Ariadne. Ariadne is built fully within the WWW infrastructure using Java and using CGI to create and access guided tour objects in a database.

3.1. The architecture of Ariadne

Figure 1 shows the architecture of Ariadne. It consists of two components in addition to the usual WWW browser and server components. The WWW servers are ordinary servers holding the WWW documents. They are accessed solely by the WWW browser when applying the Ariadne system. The browser communicates with the main controller of the system implemented in an applet. The applet is executed in its own browser window. The applet communicates with the browser using the ABP, (Ariadne-Browser-Protocol). The ABP consists of two operations:

- ShowDocument: The Applet asks the browser to show a specific document, a URL.
- GetDocument: The Applet retrieves information from the browser about the document presently being shown.[\[2\]](#)

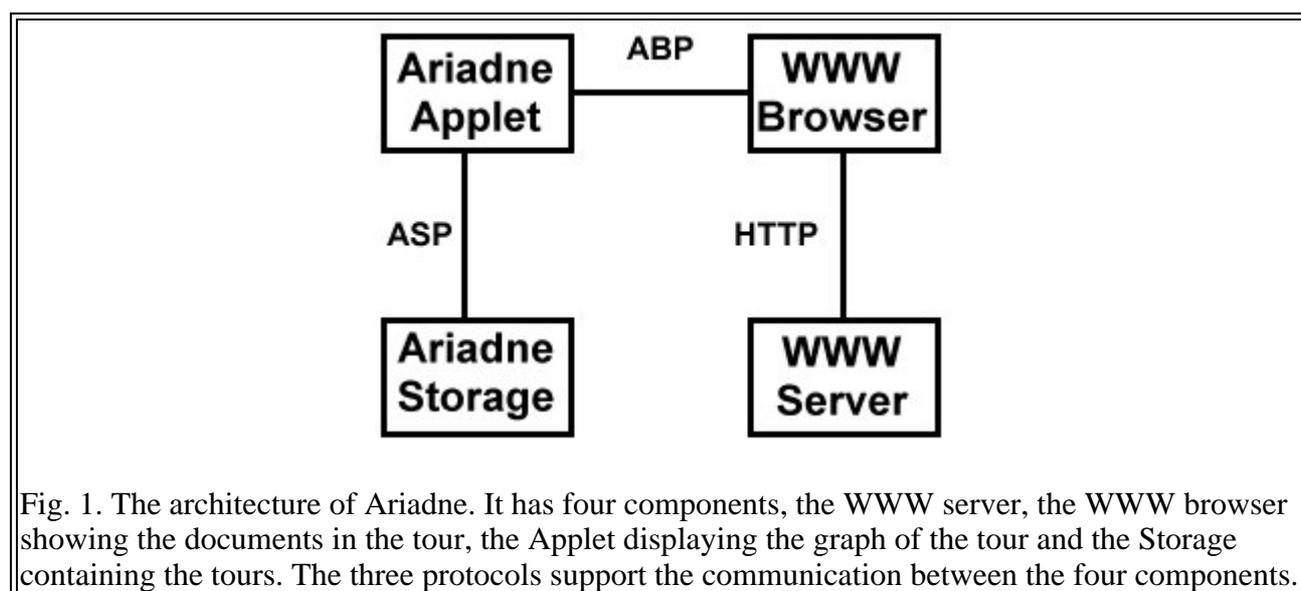


Fig. 1. The architecture of Ariadne. It has four components, the WWW server, the WWW browser showing the documents in the tour, the Applet displaying the graph of the tour and the Storage containing the tours. The three protocols support the communication between the four components.

The ShowDocument operation lets the applet control which document the browser should retrieve from which WWW server. GetDocument fetches the URL from the browser. The applet communicates with the Storage component using the ASP (Ariadne-Storage-Protocol). This protocol includes possibilities for adding, removing, updating and retrieving tours from either a database or a file system. When a tour is stored in a database, its editorial access is controlled by a password. A

user is only allowed to update and remove a tour if the password is known. The protocol uses the following operations:

- AddGuidedTour
- RemoveGuidedTour
- UpdateGuidedTour
- GetGuidedTour
- SearchTour

Both protocols provide the basic functionality, but it is possible to extend the protocols with more functionality. It would be fairly simple to remove the present backend and replace it with a more advanced database. This will among other advantages add the possibility of viewing other tours crossing the tour presently being followed, just like Bush anticipated it in his article "As We May Think" from 1945 [2]. But it will also add the possibility of constructing tours collaboratively. The current version of Ariadne is limited to storing tours in a central special purpose database. This is due to the fact that Ariadne is currently implemented as an unsigned applet, which only allows access to the site it comes from.

3.2. The user interface of Ariadne

As described in the previous section, the user interface of Ariadne is combined of two separate windows: The graph window displays the tour [3] and the browser window contains the WWW document corresponding to the current node in the graph. As can be seen in Fig. 2, the prototype of Ariadne includes a special Toolbox window with the navigational and editorial functions. This window will be integrated in the graph window in the upcoming version of Ariadne. [4]

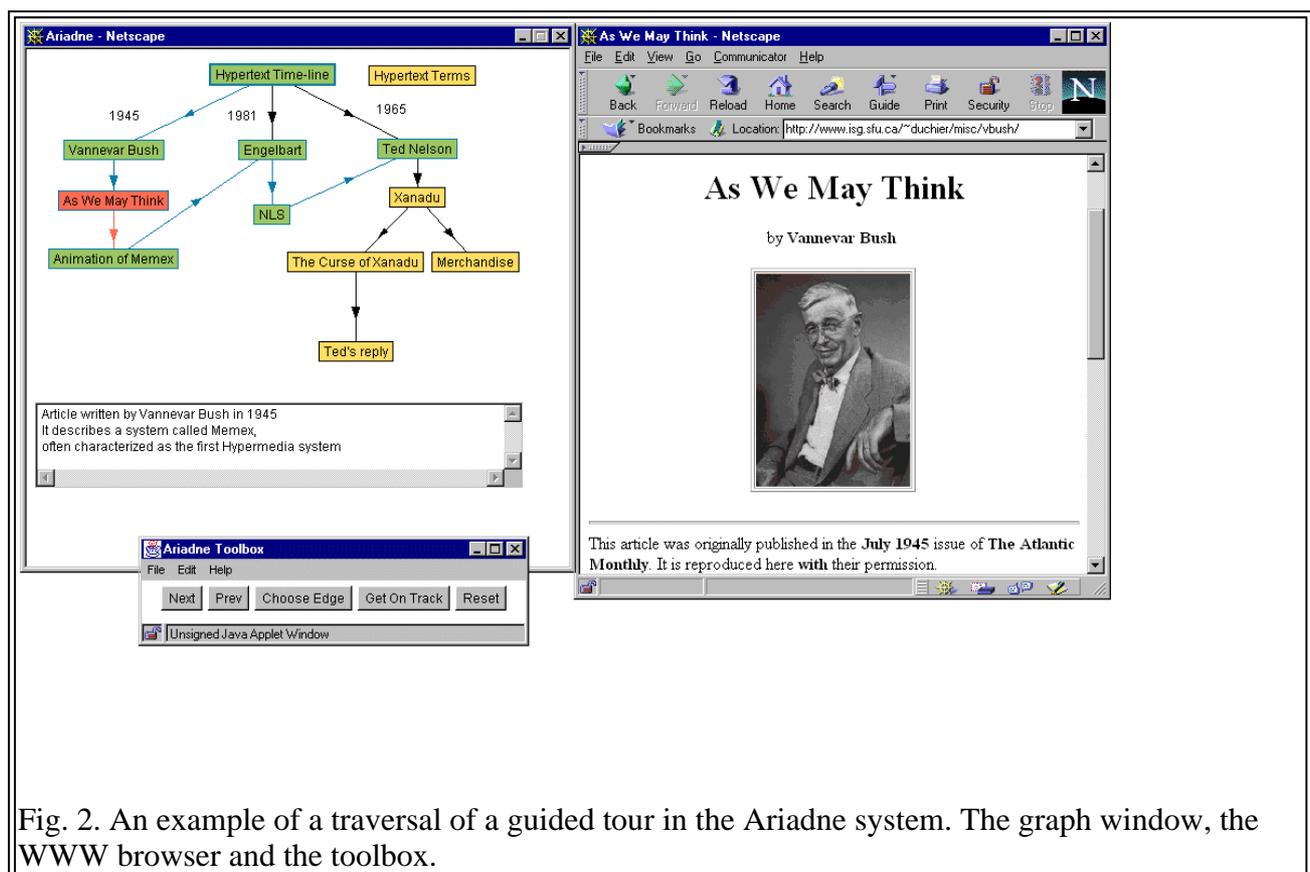


Fig. 2. An example of a traversal of a guided tour in the Ariadne system. The graph window, the WWW browser and the toolbox.

With respect to the user interface there has been a main focus on keeping the browser and the graph display separate. The boxes in the graph window highlight the names that the author has associated

to each node in the tour. The arrows connecting the boxes indicate a path between two nodes. One box is surrounded with a larger border than the others; this is the node starting the guided tour. The node currently in focus is red, unvisited nodes are orange and visited nodes are green. The arrow selected to be followed if the user activates the next button is red, black arrows represent parts of the tour not visited yet and, finally, blue arrows represent the visited parts of the tour. The graphical representation of the guided tours is inspired by [12]. The annotations of the node in focus can be seen at the bottom of the graph window. The detailed functionality of traversing and creating guided tours in the Ariadne system will be explained in the next section.

4. Usage of Ariadne

In this section, we will describe how Ariadne is used to traverse, create and update a guided tour. The main examples of usage of Ariadne are educational, where a teacher or a student wishes to present a topic being documented on the WWW in a coherent fashion with a given sequence.

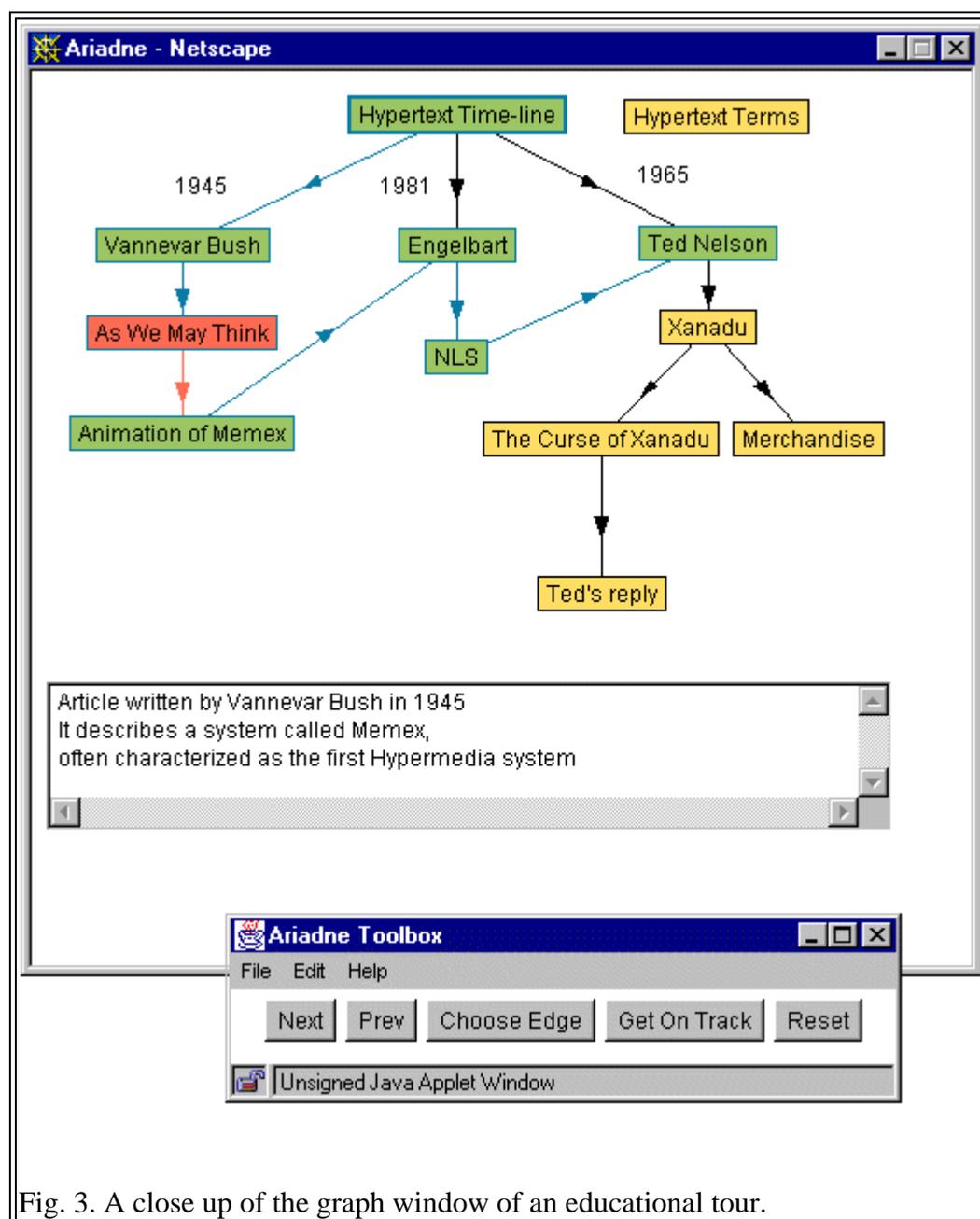


Fig. 3. A close up of the graph window of an educational tour.

Figure 3 shows an example of an Ariadne tour answering an exercise for an introductory lecture in a

Hypermedia class at Aarhus University. The student's assignment was to make a guided tour presenting the pioneers of the hypermedia field to people. The resulting guided tour presents Web pages describing Bush, Engelbart and Nelson. The tour starts with a general introduction to the hypermedia research field, then the user can choose between sub-tours on the three main pioneers in the field. First a bio sketch for the person is given and then different Web pages describing their various articles and research. Labels have been added to briefly describe the merits of the three researchers. Furthermore, the tour contains a node with no in- or outgoing edges; this should be seen as a reference page holding key information not necessarily needed in a specific place in the tour. The different colouring of the nodes indicate how the user has visited the tour. The green node has been visited, the orange is unvisited and the red node is the node in focus at the moment. If the user was to pursue further into an article by following a link in the browser window, the system would return to the page which node was red, if the Get on Track button was pressed.

4.1. Traversing a guided tour

Starting a guided tour is similar to following a normal WWW link, the user simply activates the link by using the mouse. Hidden to the user a JavaScript function is called, this function opens another browser and loads the Ariadne applet in it. If the JavaScript function is called with the identification of a guided tour as parameter, the Ariadne applet will load the specified guided tour and start executing it. The identification of a guided tour in the current version of Ariadne is a unique ID number identifying the tour in the database. On initialisation, the Ariadne applet opens a Toolbox window used for navigating the guided tour.

4.1.1. Graphical visualisation of a guided tour

When a guided tour is loaded a graph of the tour is displayed in the Ariadne applet window. The graph is a visual display of the guided tour. Every node in the guided tour is displayed as a little box containing the name of the node. Every edge in the guided tour is displayed as an arrow from its source node to its destination node. Every guided tour has a start node. The start node is the node that the tour should be initiated with. In the graph this node is indicated with a thicker border. The colour of the nodes changes from orange to green when the user has visited them. The colour of the edges changes from black to green when they have been followed. The node in focus — i.e. the node the user is reading at the moment — is red. If the node in focus has outgoing edges one of these is red — indicating that if the user wants to see the next node the edge with this colour will be traversed. This colouring of the graph is created with the intent of helping the user's navigation in the guided tour.

4.2. Navigation in Ariadne

The Ariadne toolbox window with the menubar is shown at the bottom of Fig. 2. This toolbox window is used for most navigational activities. It contains five buttons and a set of menus. The buttons are:

1. Next — follows the selected (red) edge going out of the node in focus (red) change the incoming node of the selected edge to be the new node in focus.
2. Prev — moves the user back in the guided tour.
3. Choose edge — allows the user to choose between the outgoing edges of the node in focus.
4. Get on Track — reloads the URL of the node in focus in the browser. This is used by the user to return to the guided tour, if it was left, e.g. through conventional link following.
5. Reset — restarts a guided tour marking all the nodes and edges as unvisited.

The user can use the graph of the guided tour as a means to navigate, by using the mouse to click on one of the boxes representing the nodes. When a box is clicked on, the node it represents will be the node in focus in the browser window. This functionality gives the user the opportunity to jump

between places in the guided tour.

The above functionality is similar to the ones proposed by Trigg [12]. Except for the behaviour of the Prev operation. The functionality of Prev has been designed to follow the tour backward along the ingoing edge of the node in focus. This happens even if the user jumped to the specific node. If the user came to a node by means of jumping and the node is unvisited and has more than one ingoing edge, the user gets a choice of which of the ingoing edge should be followed backwards. In other systems, e.g. [12] the Prev operation behaves like the Back operation in a WWW browser and returns the user to the place where s/he came from before executing the jump. We have chosen this approach because it seems natural when using a real world guided tour in museums and cities as a metaphor. Using a book, as the metaphor would lead to the same conclusion on Prev: the turning of pages in a book could easily be compared to the use of Next and Prev in a guided tour. If you jump ahead in the book the behaviour of Prev should not be jumping back to where you came from in the book. It would be more logical if you turned to the page preceding the page you were looking at. If the reader wanted to return to the previous location in the tour this could be obtained by jumping back in the tour, just like returning to a previous location in a book.

4.3. Authoring guided tours

A user can load and save guided tours via the File menu in the Toolbox window. Activation of the Save operation prompts the user for the password coupled to the specific guided tour before storing of the tour is allowed. It is possible for a user to search databases containing guided tours. This is done by activation of the Search option in the File menu of the Toolbox window. The user can indicate a substring of the creator and/or a substring of the guided tour name in a form and the system will return a list of the guided tours matching these properties.

Editing Ariadne guided tours is enabled with the Shift key. Holding down the shift key brings Ariadne to edit mode. In edit mode a node can be selected by double clicking on it. When a node is selected it can be deleted, changed to be the new startnode or its properties can be changed. This is all done by operations in the Edit menu. If a node is selected and the user clicks on another node an edge starting from the selected node and ending in the other node is created. If an edge already exists, it is overwritten. If the user wants to change a node's position in the graphical representation of the guided tour, the user should hold down the shift key then drag and drop the node with the mouse. The arrows are automatically adjusted. If the user wants to add a new node to the guided tour the New Node option in the Edit Menu should be selected. This will bring up a dialog window in which the user can type in or cut and paste the name, URL and annotation of the new node. When the dialog OK button is pressed the new node can be placed in the Ariadne window with the mouse. Labels with explanatory text can be created and edited in the same way as nodes.

5. Use experience with Ariadne

We have accomplished a user evaluation of Ariadne by soliciting answers to a questionnaire from users who have had the opportunity to try out Ariadne. Ariadne has been made available to the public through announcements on different relevant newsgroups. Furthermore, we involved a number of students and graduates from the Faculty of Science, University of Aarhus, Denmark in a more qualitative evaluation. After having tried out the system we used a qualitative interview method to get the users' reaction to the system. The evaluation revealed the following strengths and weaknesses of the system:

Four strengths were emphasised in many of the responses to questionnaire and interview:

1. Almost all the evaluators explicitly emphasised the graphical representation of the tour as a help to maintain their overview of the tour.

2. The evaluators were positive towards using Ariadne as a presentation tool for coherent topics.
3. The users felt instantly reminded of where they came from when going off a guided tour from an interesting page. Some of the users also used the get-on-track button at some point to get back on the tour after a sidetrack.
4. All the users found branching tours useful, in particular the idea of combining sub-tours into larger tours.

A number of weaknesses in the current design were also uncovered:

1. Some evaluators were sceptical as to whether a very large tour could be handled satisfactorily in the graphical display.
2. Some evaluators found that a high number of branches was confusing
3. Some evaluators found it inconvenient to have both a graph window and a toolbox window, because they together took up too much space on the screen.

Finally, a few issues for further research on the design of the system were raised.

1. How do we avoid the graphical layout being too large and confusing?
2. How do we avoid the number of branches in a tour becoming too large and confusing?

This is what we could call a scaling issue, how should we manage large and complex guided tours. These problems must mainly be handled by the guided tour authors, through modularization of tour and careful design of the layout. The graphical layout of the tour is manually constructed by the tour author. However, the problem of large graphs can be handled by adding an overview facility to the graph window similar to what is done in NoteCards browsers [7]. This would allow the user to view a miniature view of the tour and select a sub area of the tour to view in full size.

To fully clarify the problems related to large tours, we acknowledge the need for a large-scale user evaluation.

6. Ariadne compared to existing guided tour systems

In this section, we make a more detailed comparison of Ariadne with the three main existing guided tour systems for the WWW.

The systems Footsteps [5], PGTS [8] and Walden's Paths [11], which we have examined, all use the same technique to presenting the guided tours. A guided tour server contains the information on all the tours created. When the user wishes to follow a guided tour, the tour server is contacted; the tour server retrieves the first document in the tour from a WWW server. But before returning it to the user's WWW browser, it is pre-processed with a number of additions to the HTML code. The tour server does the following:

- Adds navigation graphics (all three systems);
- Changes links in the document so they go through the tour server (Footsteps and Walden's Paths);
- Adds annotations (PGTS and Walden's Paths).

This approach has a number of disadvantages:

- Increased Internet traffic — all documents are retrieved first by the tour server then in turn sent to the user's client.
- Pre-processing limits the browser's navigational options: Bookmarks, Home, Open Location etc. will not access the tour server. Use of these operations will terminate the traversing of the

tour.

- Pre-processing is not an easy task — None of the systems can handle JavaScript; Walden's Paths can not handle Forms; finally, none of the systems can support frames.
- The editor and the viewer of a tour are two separate tools. All three systems have a separate editor for creating guided tours.
- Lack of overview of the tour — None of the systems give a satisfactory presentation of an overview of the nodes included in the tour.
- Only linear tours are supported.

The pre-processing approach has a number of advantages compared to the current version of Ariadne:

- It is possible to bookmark a page in a tour and return to this page within the context of the tour later.
- The other systems occupy less screen space for presenting their guided tours.

We are currently implementing Ariadne solutions to these issues, and we claim that the approach we have taken to guided tours represents an improvement over the existing systems. The user of Ariadne gets a better overview of the tour, because the tour is visualised in parallel with the WWW browser containing the documents presented. Moreover we have overcome the problems with the pre-processing approach, since no pre-processing is needed. This implies that we can make pages in a guided tour appear in the same way as if viewed with the WWW browsers.

7. Conclusion and future work

This paper has presented a new Java-based guided tour system, called Ariadne, for the WWW. Ariadne represents an improvement over existing guided tour systems in the following areas: (1) independent user interface which does not affect the layout of the documents being part of the tour, (2) branching tours where the user may follow alternative routes, (3) composition of existing tours into aggregate tours, (4) overview map with indication of which parts of a tour have been visited and support for getting back on track.

The Ariadne prototype is being further developed in the Coconut project between University of Aarhus and TeleDanmark Internet. One of the approaches taken to improve the interface and functionality is to make the Ariadne Applet signed. This enables the following enhancements to Ariadne:

- The applet is able to request both the URL *and* the name of the document the browser is showing with the GetDocument operation. This makes the process of adding new nodes to the tour more elegant, e.g. addition of the document presently being viewed in the browser to the tour by a simple press of a button.
- Tours can optionally be stored on the user's own hard disk or on an arbitrary server chosen by the user (providing s/he has write access).
- A node in a tour can contain another tour and not just a WWW document. This feature is rather simple to add due to the open backend of our system.

A second goal is to use the Devise Hypermedia framework [6] as a backend to Ariadne. This integration with a full fledged open hypermedia engine will give us a number of value adding structuring mechanisms as well as collaboration support for guided tour building.

A third goal is to add a command language. In [9] we outlined how this could be done with a small event based language. Our language was based on the ideas outlined by Zellweger in [13] combined with an intuitive and visual programming style.

An example of features provided with this language, is conditional guided tours. For instance, a guided tour author can hide certain parts of a guided tour until the student has visited specified nodes. The teacher simply specifies this with an attribute on the nodes making them hidden until the student has triggered an event by visiting a specific node.

Acknowledgements

This work has been supported by CIT (The Danish National Centre for IT Research): grant no.123. We would like to thank the evaluators for supporting us with answers to the evaluation.

Notes

{1}: In Greek mythology Ariadne was the daughter of King Minos of Crete. Theseus — the Prince of Athens — found his way out of the Labyrinth of the Minotaur with the help of a ball of twine given to him by Ariadne. We have named our system Ariadne, because we view guided tours as the thread that can help the users when exploring the Labyrinth of the WWW.

{2}: Due to technical problems originating from the Java security system, this operation requires the applet to be a signed applet.

{3}: The code used for displaying the graph is based on the code example from Sun called GraphLayout.

{4}: The Toolbox window is a stand-alone window and not included in the Ariadne applet, since it was not possible to add menus to an applet at the time of development.

References

[1] T. Berners-Lee, R. Cailliau, J. Groff, World Wide Web: the information universe, *Electronic Networking Research, Application and Policy*, 1(2): 52–58, 1992.

[2] V. Bush, As We May Think, *The Atlantic Monthly*, 76: 641–649, August 1945.

[3] V. Bush, Memex II, 1959, in: J. Nyce and P. Kahn, (Eds.), *From Memex to Hypertext: Vannevar Bush and the Mind's Machine*. Academic Press, 1991.

[4] L. Carr, D. De Roure, W. Hall and G. Hill, The distributed link service: a tool for publishers, authors and readers, in: *Proc. 4th International World Wide Web Conference: The Web Revolution*, Boston, Massachusetts, USA, 1995.

[5] R. Furuta, F.M. Shipman III, C.C. Marshall, D. Brenner and H.-W. Hsieh, Hypertext paths and the World Wide Web: experience with Walden's Paths, in: *Hypertext '97 Proceedings ACM*, 1997, pp. 167–176.

[6] K. Grønbaek, N.O. Bouvin and L. Sloth, Designing Dexter-based hypermedia services for the World Wide Web, in: *Hypertext '97 Proceedings ACM*, 1997, pp. 146–156.

[7] F.G. Halasz, T.P. Moran, and R.H. Trigg, NoteCards in a nutshell, in: *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, 1987, pp. 45–52.

[8] <http://www.infosys.tuwien.ac.at/GuidedTour/GuidedTour.html>

[9] A. Jensen and J. Jühne, Udvikling af en stimekanisme til WWW, Department of Computer Science, University of Aarhus, 1997.

[10] H. Maurer (Ed.), *HyperWave: The Next Generation Web Solution*. Addison-Wesley, Harlow, 1996.

[11] D. Nicol, C. Smeaton and A.F. Slater, Footsteps: trailblazing the Web, in: *Proc. of the 3rd International World Wide Web Conference*, Darmstad, 1995.

[12] R.H. Trigg, Guided tours and tabletop: tools for communicating in a hypertext environment, *ACM Trans. Office Information Systems* 6(4, October): 398–414, 1988.

[13] P.T. Zellweger, Scripted documents: a hypermedia path mechanism, in: *Proc. ACM Hypertext '89*, Pittsburgh, Pennsylvania, 1989, pp. 1–14.

Vitae



Jesper Jühne is system developer at Tele Danmark Internet, Denmark. He is currently working at the Coconut project, a co-project with the Dept. of Computer Science, University of Aarhus, Denmark. He finished his master's degree in 1997 from the Dept. of Computer Science, University of Aarhus. His research interests are: World Wide Web; Hypermedia and Multimedia; Computer Supported Cooperative Work and System Development.

Anders Thorbjørn Jensen is a Technical Producer at LEGO A/S, SPU-Darwin, Denmark. He is currently working on a CD-ROM production involving Robotics. He finished his master's degree in 1997 from the Department of Computer Science, University of Aarhus. His interests are: Robotics, World Wide Web and Hypermedia.



Kaj Grønbæk is associate professor at the Department of Computer Science, University of Aarhus, Denmark. He finished his master's degree in 1988 and his Ph.D. in 1991 both from the Dept. of Computer Science, University of Aarhus. His research interests are: Hypermedia; Multimedia; Computer Supported Cooperative Work (CSCW); Cooperative Design (system development with active user involvement, cooperative prototyping); User interface design; object oriented tools and techniques for system development.