The Help-Desk Assistant Project

Peter Clark
Dept. Computer Science
Univ. Texas at Austin
Austin, TX 78712, USA
pclark@cs.utexas.edu

1 Project Summary

The DCE Help-Desk project aims to leverage knowledge-base technology to usefully and profitably augment existing customer services. The goal of the project is to build a help-desk assistant, capable of automatically answering a proportion of customer questions which would otherwise have been phoned in to a normal help-desk. The Help-Desk Assistant complements existing technologies for customer support – in particular **on-line manuals** and **databases** of example problems+solutions – by generating answers on-the-fly from a structured representation (the **knowledge base**) about the domain, rather than reciting opaque paragraphs of text. By using a structured representation, the Assistant can customize answers to the end-user's specific situation and level of expertise, thus enabling end-users to self-answer some of their own questions rather than phone in a query. The challenge with this technology is to overcome the high cost of building such representations in the first place. We are addressing that challenge with a component-based approach to knowledge base construction. Our resulting prototype suggests this technology can be applied for customer support in a financially profitable way.

As the application goal, we are constructing a prototype system capable of answering novice users' questions about a particular sub-domain of computing, namely DCE (OSF's implementation of a distributed computing environment). Customers' questions are answered directly through a WWW interface to the question-answering software. As the research goal, we are developing better methods for constructing such knowledge bases, based on assembling them from a library of reusable components. Our research includes building such a library, to serve as a resource for this and future knowledge-based projects.

This project has a two-year duration, and is sponsored by the Digital (Customer Services Division) and the Texas Advanced Technology Program. It is being conducted by the Knowledge-Based Systems Group at Univ. Texas at Austin. The project began mid-1994.

2 Research Foundation

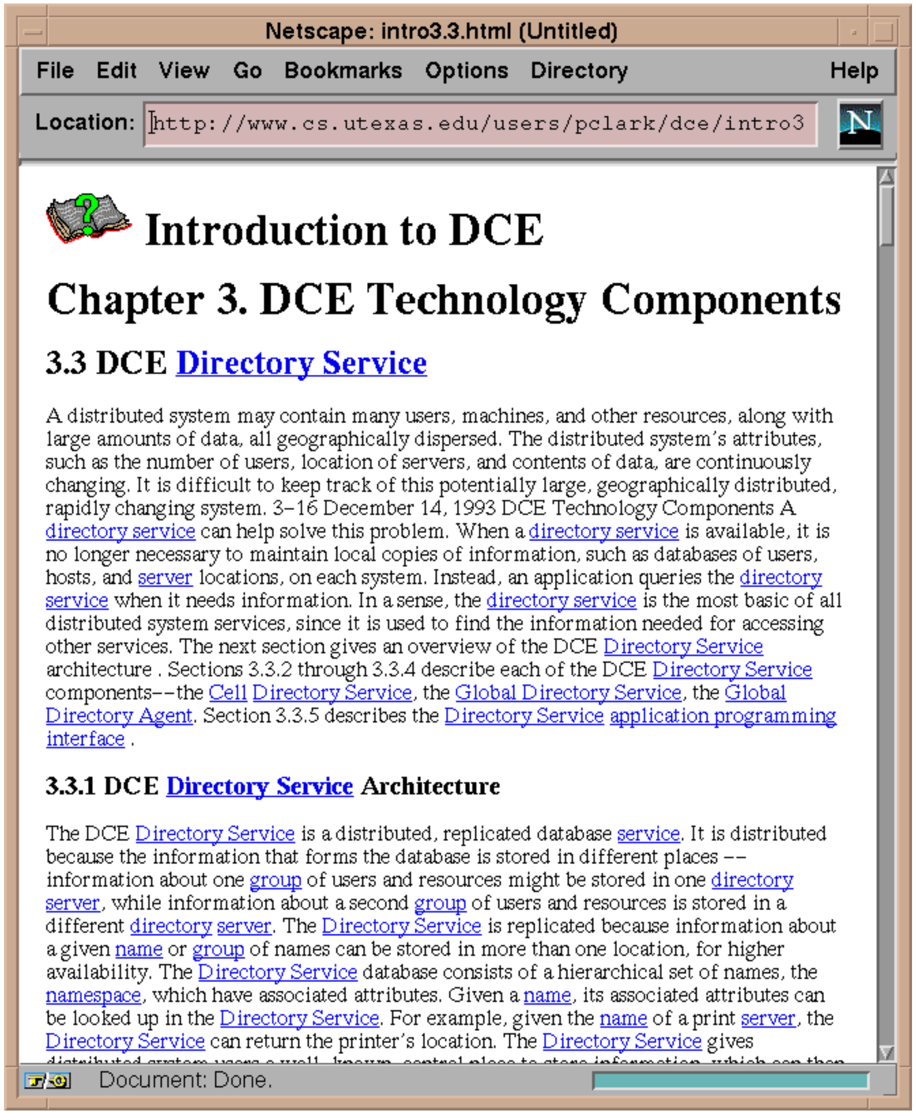
The knowledge base contains information about computing concepts and their relationships, which are used to infer answers to users' questions. This information is represented as a richly connected network of symbols, rather than blocks of text whose meanings are inaccessible to the computer. The network has a clear logic-based semantics, and is expressed as a frame hierarchy for efficient reasoning. With information in this form, the Assistant can infer answers to users' questions at run-time, interacting with the user as needed to acquire additional information, rather rather than simply recite pre-built "canned answers". This allows it to customize its answers to the user, control the level of detail, and answer questions which were unanticipated by the system designers when it was constructed, as illustrated in the following pages.

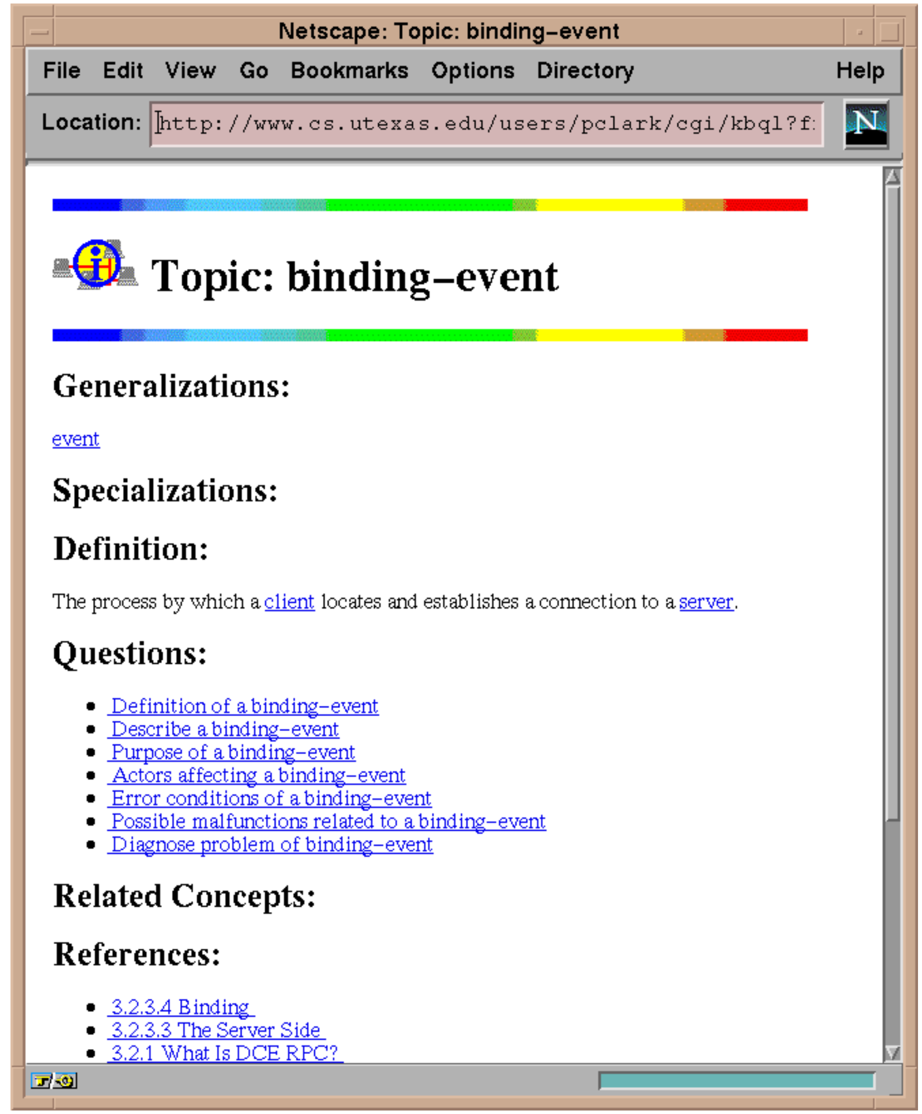
Most importantly, we are constructing a **library of representational components** from which this, and other, knowledge-bases can be easily constructed. Computing-specific concepts are built by combining components in the library together. (For example: a representation of a 'database' can be built from components modeling a 'container', a 'secure resource', and a 'service provider'.) This 'knowledge library' has a role analagous to a software library, and is intended as a key, long-term resource for constructing this and other such systems. Given the complexity of modern software systems, and explosion of on-line documentation, we envisage inference-capable, interactive, question-answering systems of this kind will become increasingly important for making it easier for people to understand and use computers.

3 The Application Software

The Assistant's knowledge base is about DCE (OSF's implementation of a distributed computing environment). Similar techniques could be used to encode knowledge about other products, artifacts or general subject domains.

The following pages illustrate the current prototype in use, for help about DCE. They show how the Assistant can infer customized answers to user questions, and convert and present those answers as natural language text.



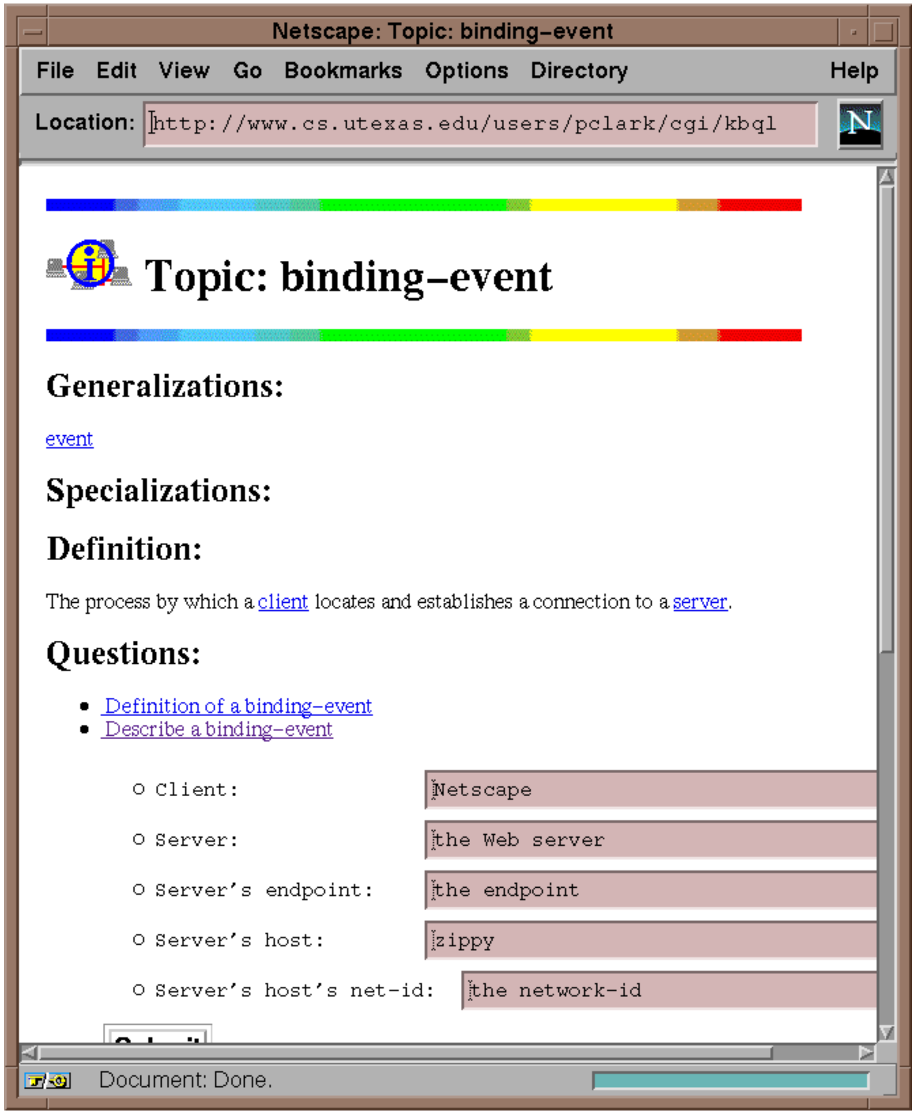


This first screendump shows an on-line manual page, which the user is viewing. This is just like a normal manual page, except the Assistant has automatically added hyperlinks (shown in blue) to concepts which are in its knowledge-base.

If the user wants to find out more about one of these concepts, he/she clicks on it. This sends a signal to the Assistant to synthesize a page about that concept from the information in its knowledge-base, as illustrated in the next screendump.

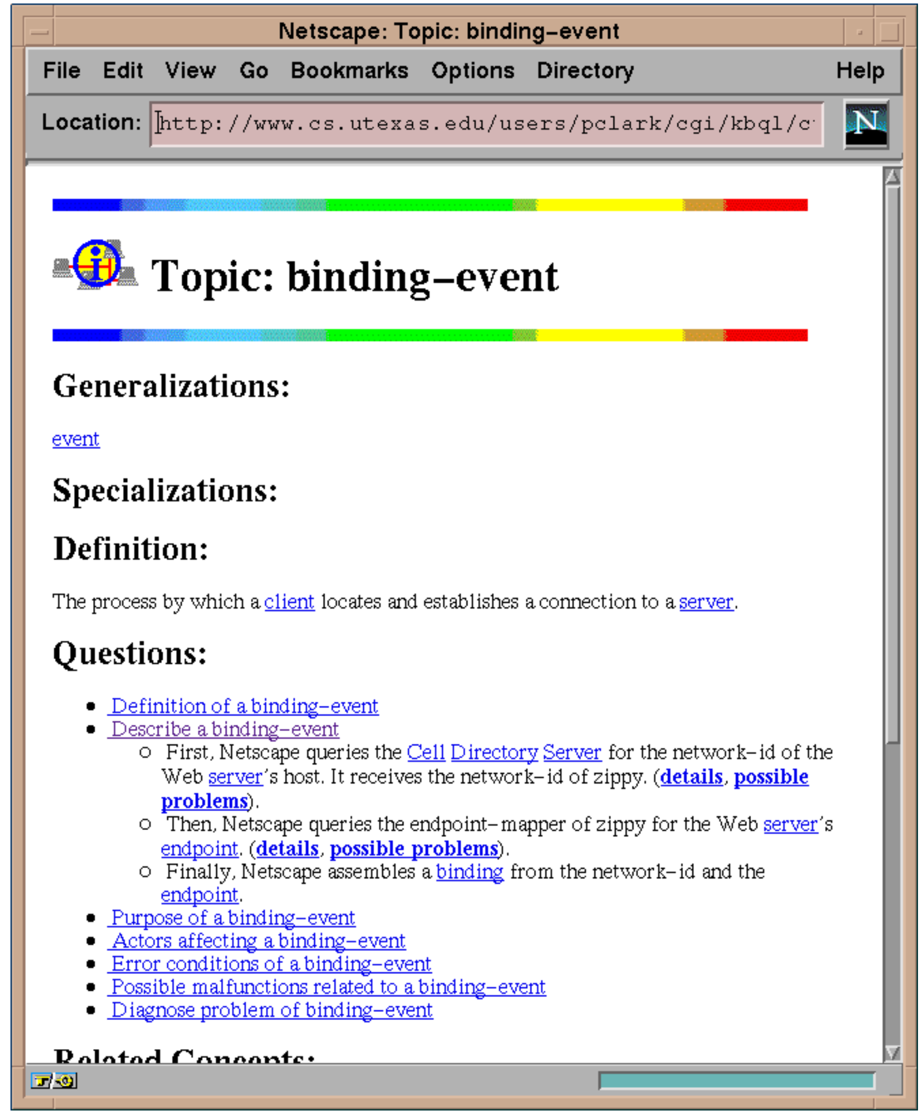
The user clicked on `binding`, and the Assistant produced this page. Note that it is generated at run-time, using information in the Assistant's knowledge-base, rather than being a pre-stored document. For this topic (binding), it shows: its generalizations, its definition (pre-stored text from a glossary), and references back to other on-line pages about binding. These allow the user to navigate to information about other concepts, or to other manual pages.

Additionally, and most importantly, the Assistant presents a list of questions which make sense for objects of this type (events). To ask one of these questions, the user clicks on it, causing the Assistant to interactively generate an answer to that question.



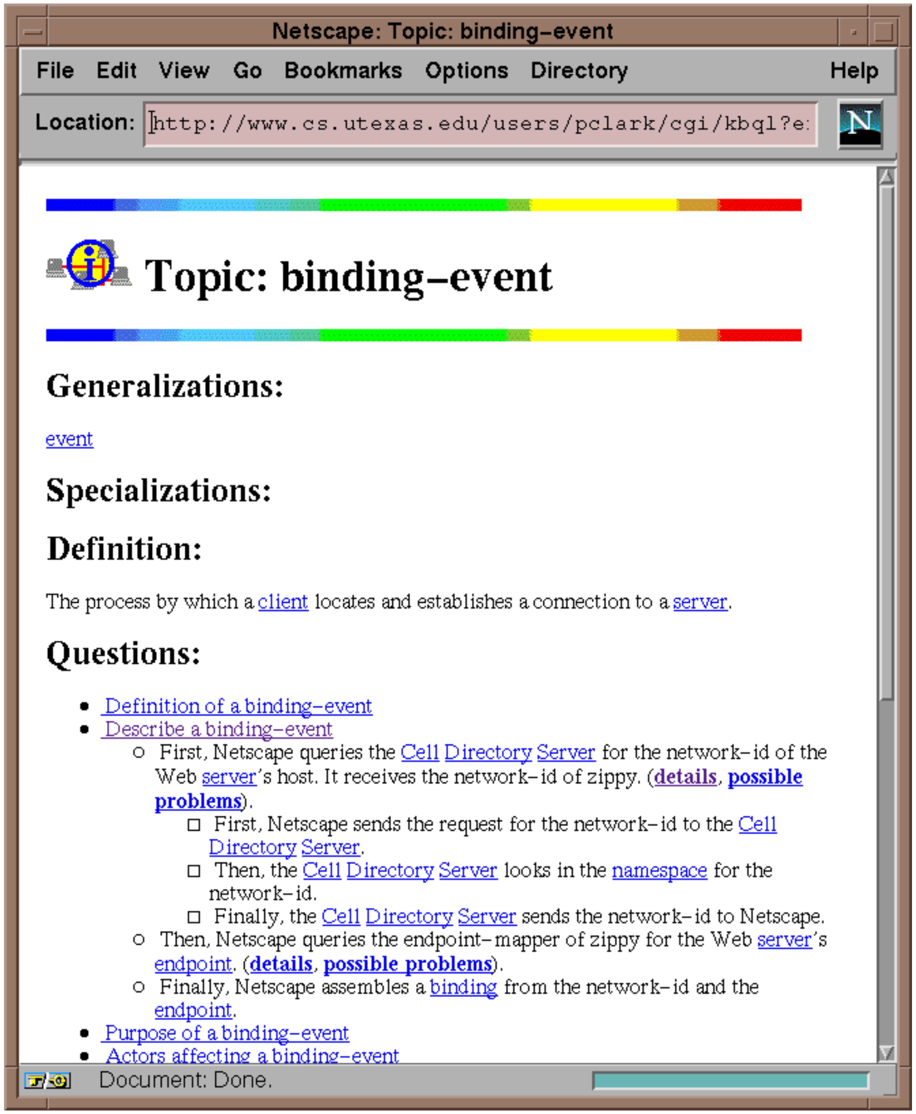
I: Description Generation. The user clicked on `Describe binding event`. To answer this question, the Assistant (i) builds a representation of 'binding' using information in its knowledge base (ii) asks the user to customize the representation (iii) extracts information required for the text description (iv) convert the extracted information into natural language text.

Here stage (ii) in this process is shown, where the user is asked to customize the representation by giving names to selected objects which are involved. The user has customized the client to be `Netscape`, the server to be `the Web server`, and the server's host to be `zippy`. (Again, note this page is machine-generated at run-time, not a pre-stored document).

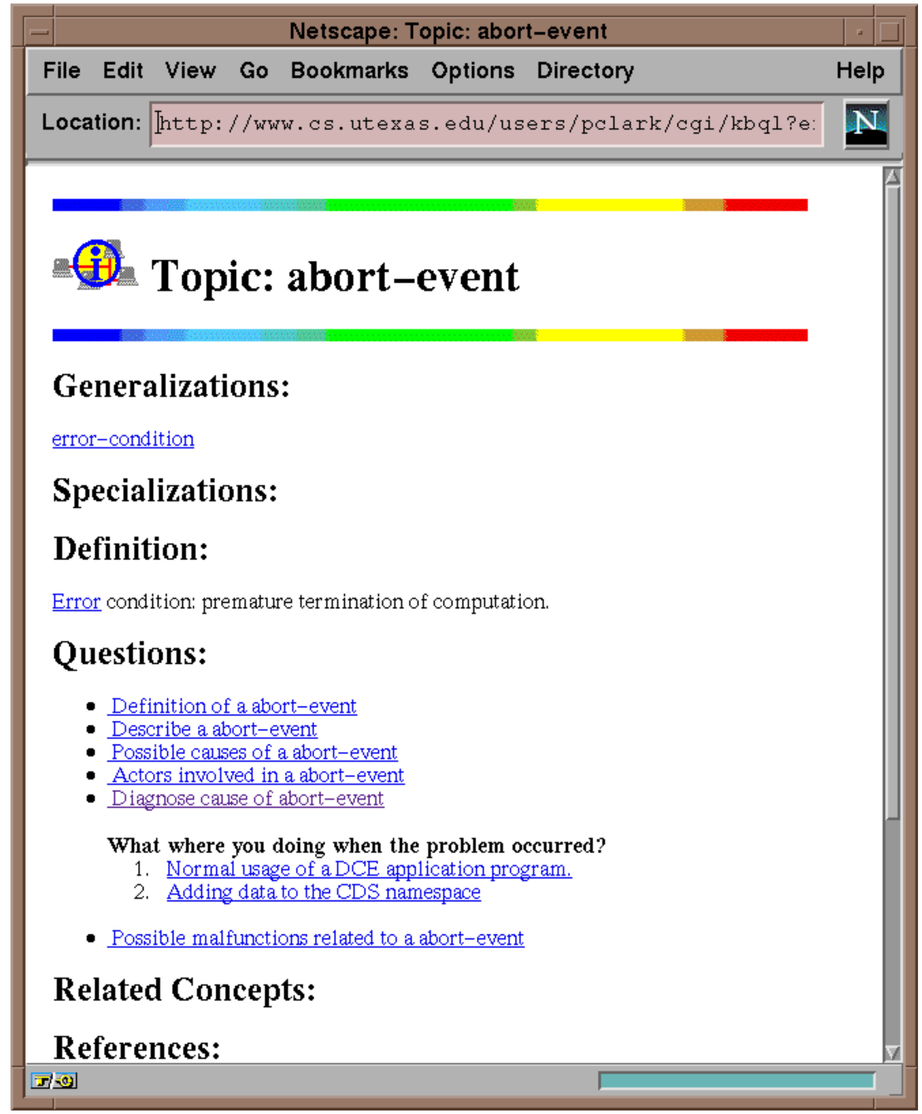


Having built a representation of 'binding', customized to the user's situation, the system converts it to natural language text and presents it to the user, as shown above.

From here, the user can ask for more details, or ask another question, move to another concept, or move back to the manual pages.

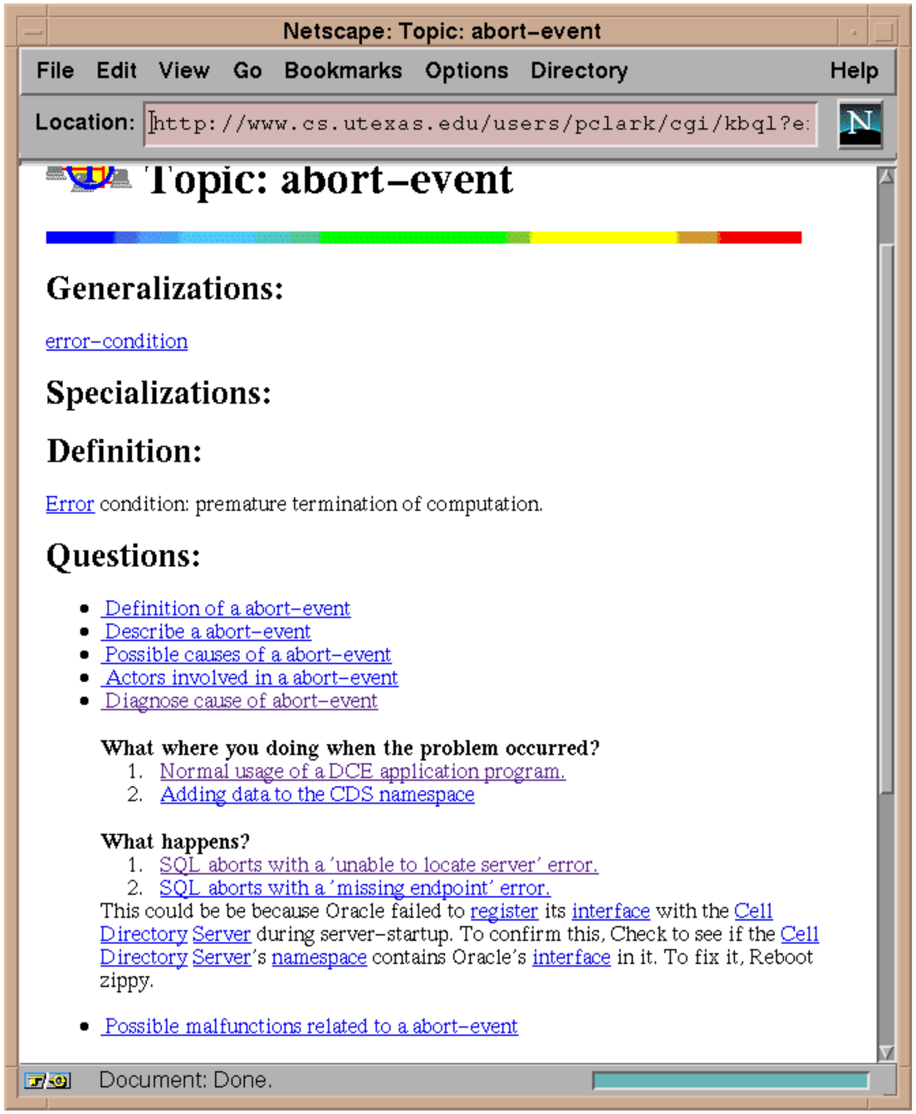


Following on from the previous screen, here the user has clicked on `details` for the first subevent in binding. The Assistant has then constructed a more detailed representation of this first step from information in the knowledge-base, showing the three subevents of this step, and again converted the description to text.



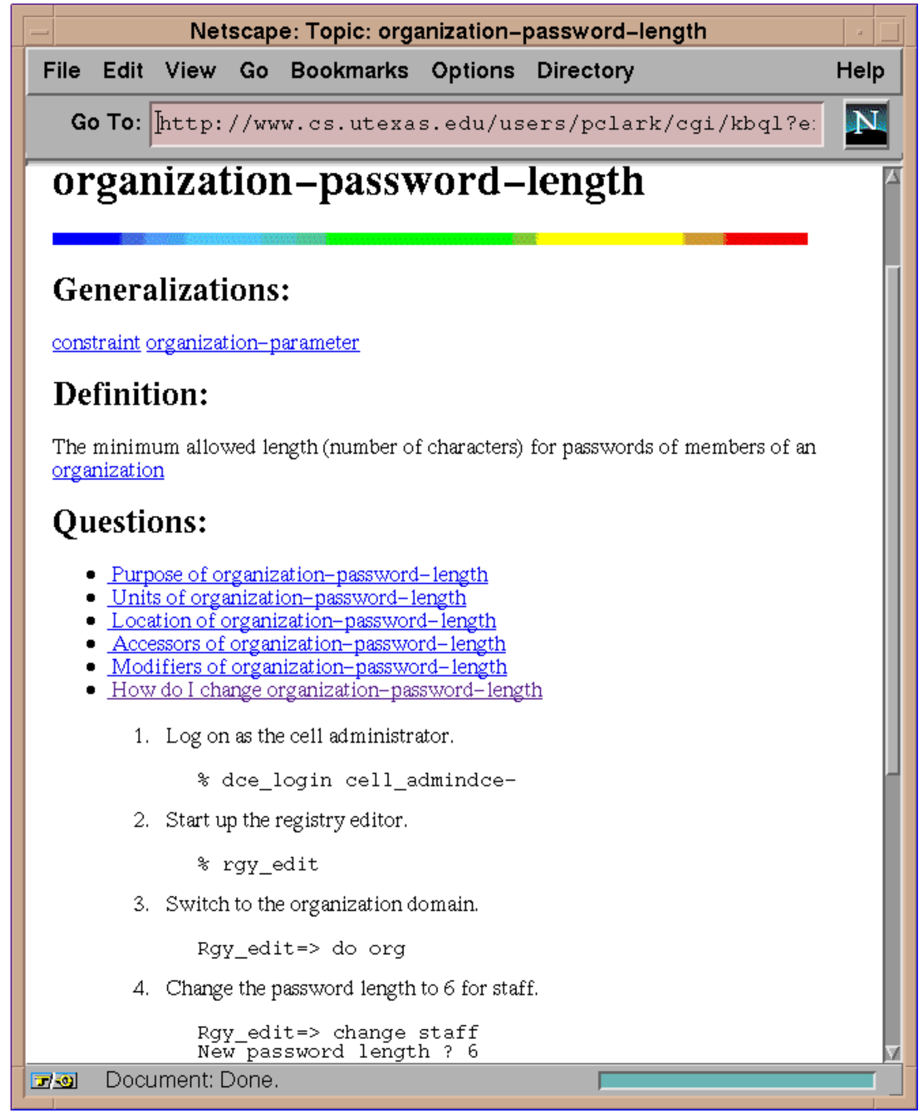
II: Diagnosis. This screen illustrates simple, interactive diagnosis by the Assistant. The user has moved to look at a different concept, namely `abort-event`, and has clicked on the question `Diagnose cause of abort-event`. The system will now ask the user a series of questions in order to home in on what the problem is, following a simple diagnostic strategy.

Here the system is asking the first question, namely what class of activity was the user involved in when the problem occured. The user will click on the appropriate answer.



This is a continuation from the previous screen, after several more questions have been asked (including questions to customize the answer, here to an SQL-Oracle interaction). During the interaction, the Assistant is building up a representation of what the user was doing and what might be the problem.

After homing in on possible diagnoses, the system again converts its internal representation to text and displays it to the user. The text here suggests a cause of the user's problem, confirmatory tests, and possible fixes.



III: Planning. This screen illustrates the Assistant using a planning algorithm to answer a "How do I...?" question. The user is viewing the concept `organization-password-length`, and has clicked on the question `How do I change the organization-password-length?`.

To answer this question, the Assistant tries to find an action in its knowledge-base which will achieve the user's goal. If one is found, it then looks for actions which will achieve any preconditions of the first action, recursively until all the conditions for performing the action are satisfied. The resulting plan is then converted to natural language, and displayed to the user as shown above.