Stanford DLITE User Study

Mark Mortensen
8/20/97

Abstract:

User tests were conducted on the DLITE digital workspace. These consisted of observed use of the DLITE system followed by an interview by the test administrator. The tests themselves were carried out both remotely over a network and locally in the Digital Libraries Lab, on subjects with moderate computer knowledge. Initial tests resulted in system failures that caused DLITE to crash or become totally unusable. In subsequent tests, users noted a number of areas of DLITE that caused confusion, in particular: the instantiation of queries, the purpose and functioning of the graphics in the upper left corner of objects, and the obscuring of objects in the workspace when dragging large components. Given the reactions of users in the post-test interview, these problems do not appear to be flaws in the design of DLITE, but implementation errors not intrinsic to the model upon which the functionality is based.

Introduction:

User testing of the DLITE digital workspace system was carried out between February and June of 1997. These tests were designed to indicate those aspects of the system that were problematic to users. This has a dual purpose: to better understand the system for future redesigns, and to make some adjustments to it along the way. Before addressing these issues, however, it is important to understand the design model upon which it was based.

The DLITE digital workspace is an interface for document / service interaction designed by Steve Cousins at Stanford University. The workspace relies upon graphical instantiations of all documents and services and uses drag-and-drop manipulation to accomplish tasks. The model upon which the system's functioning was designed is that dragging a document or collection of documents onto a service invokes the service on those documents. The workspace itself is customizable and reusable, allowing users to create specialized work templates for repeated tasks.

Subjects:

In selecting a group of subject for the user tests, we hoped to get as wide a range of users as possible. This includes both heavy and light computer users from various fields. Unfortunately, at the time of the user tests, DLITE was only able to access a relatively small set of commercial databases, all of which held computer-related information. As a result, the only way that we could get an accurate impression of users using DLITE to integrate numerous services was by making sure that the information the test subjects were searching for was in some way computer related. For this reason, we restricted our test subjects to individuals with some knowledge of computer science, and thus topics on which DLITE was capable of providing some information.

<u>Stanford Math and Computer Science Library</u>

As initially designed, the user tests of DLITE were to take place in the Stanford University Math and Computer Science library. A computer running Windows 95 was placed in the library to provide a dedicated workstation for the user tests. The machine was set up with a shortcut to the DLITE home page, and Librarians were trained in how to assist users of DLITE with any simple questions they might have. These users were to be library patrons with Computer-based reference questions whom the library staff steered towards using DLITE as a reference.

It was hoped that this group of "walk-up" users would meet the criteria needed for the user study. First, being in the Math and Computer Science library, their queries were more likely to be computer-based than they would if they originated from a random sample of people. Second, as the users had gone to the library for answers to their queries, it was hoped these would be representative of the types of queries DLITE would face if released. This would provide a more accurate test of DLITE than would queries thought up in a manufactured testing situation. Third, being composed of walk-up users, it was hoped that these users would form a relatively random sample in terms of computer use.

Testing in the Math and Computer Science library ran into unforeseen complications arising from an almost non-existent subject pool. After having been deployed for two weeks, there had been no users of the system. Meeting with Library staff to discuss the issue, it was brought to our attention that the number of patrons of the Math and Computer Science Library with computer-focused questions had decreased significantly in the past year. Library staff attributed this change to the relocation of the Computer Science department to the new Gates Computer Science Building. Given this reduction in library usage, it was decided to shift the focus of testing from walk-up users to users we selected.

<u>Remote / Laboratory Users</u>

Two distinct sets of tests of selected subjects were carried out. The first set of these were carried out remotely in the subjects' own workplaces in an effort to test subjects in an environment as realistic as possible. These tests were carried out on personal computers all running Windows95 and connected to the network via 10BaseT Ethernet connections. The second set of tests was then carried out in a controlled laboratory environment, within the Digital Libraries Laboratory in the Gates Computer Science Building. These tests were carried out both on a PC running Windows95, as well as a HP workstation running HP-UX.

The users selected for both sets of tests were individuals comfortable with computer use and with relatively high levels of computer knowledge, in order to allow them to generate queries to the Computer-Science related databases which would be consistent with the expected use of the system. From this point on, all references to user tests refer to those carried out either remotely or in the Stanford Digital Libraries Laboratory.

Due to problems that will be explained at greater length below, a number of the user tests ran into system failures that prevented them from being completed. As a result, the conclusions drawn in this study are based on a combination of partial results and completed user tests. Of the total fifteen users studied, the first four suffered these total system failures, resulting in only partial results. The next four ran into other significant system problems that significantly affected the outcomes of their tests, in some cases yielding only partial results. The final six users completed successful tests without significant system problems, yielding complete results.

Materials/Methods:

In both the remote and laboratory tests, subjects began having been told only that they were testing a new computer work environment. After the subjects filled in the information required to set up a new user account, DLITE then provided them with an introductory page about the system, as well as a set of instructions walking the subject through a simple initial task (see Appendix 2). The user was instructed that he or she could start using the system by running through the initial task, or just use it as a reference while experimenting on his or her own. Subjects were not required to run through the initial task if they did not wish to. This allowed for differences in users' preferred learning styles, where some users might not want to spend their time being led through a dummy task, preferring instead to explore on their own.

As subjects used the system, they were encouraged to "think out loud", voicing any questions or problems they came across. The administrator of the user test observed the actions of the users, and interacted with him or her only when necessary or explicitly requested in order to avoid inadvertently biasing the subject. Apart from these few interactions, the administrator was merely an observer, noting the problems the users had, and what caused them.

Users were instructed to use DLITE for however long they felt they needed in order to get a good understanding of the system and its functionality. This typically took approximately twenty minutes, after which time the subjects were provided an on-line questionnaire, which asked them a number of questions both about the DLITE interface and their user experience as a whole, (see Appendix 3). This was followed by a five-minute talk with the test administrator to gather any final opinions about DLITE. This discussion was then transcribed, and appended to the administrator's observations during the test.

User Test Results:

The problems identified by the DLITE user tests fall into two distinct categories. The first is that of "bugs", problems within the functionality of DLITE which were not part of the actual DLITE design. These resulted from programming errors, often in the form of miscommunication between the user interface and the back-end of the system. The second are "design flaws", aspects of DLITE which were designed into the system, but which user tests indicated were unclear or confusing.

It should be noted that development and revision of DLITE continued throughout the duration of the user testing. As a result, a number of the bugs and design flaws which were reported in the earlier tests were fixed by the time later tests were performed.

Bugs

Early in the testing of DLITE, two irrecoverable bugs were encountered which each made the system unusable by the test subjects. These bugs are differentiated from other bugs encountered, as these were unavoidable by a user trying to run even a single simple query.

The first of these problems involved DLITE ceasing to respond to user requests, and becoming completely non-reactive, (apart from menus and buttons which were run locally). Arising when users submitted queries, once this situation had occurred, DLITE was completely unrecoverable apart from quitting the Netscape browser, and restarting it. A sequence of tests was run on the system on six different platforms seeking to reproduce the bug. These tests indicated that a problem in the threading synchronization was causing the system to enter a state of deadlock. This bug was fixed, and user testing was able to continue.

The second problem arose in the tests run remotely, which suffered from severe latency problems making DLITE largely unusable. One common instantiation of this problem was observed when subjects attempted to use the mouse to drag objects within the workspace. In many of the remote tests, the objects they selected and dragged did not move with the cursor. This misled subjects into thinking that the action had not been registered, and then to repeat their attempts to perform the same action over again. As the actions had been registered, and were queued awaiting display, when they were displayed, the subjects' repetitions of their actions often performed a number of actions they did not expect to occur.

An initial hypothesis for the cause of these problems was network latency, causing a lag in the reaction time of the graphical interface components of DLITE so significant that the noted problems arose. Numerous uses of DLITE both across Stanford's network as well as from outside Stanford's network, however, were successful both before and after these tests were carried out. This indicated that while network lag might be a contributing factor, it was not the sole cause of the problem. It has since been hypothesized that the problems arose from the configurations of the individual remote machines not providing the resources needed for DLITE to run smoothly. This problem forced the remainder of the user tests to be carried out in the Digital Libraries Laboratory, on machines whose configuration was known to work. Once these two problems had been dealt with, no further system problems occurred.

Apart from the two above-noted unavoidable bugs, two additional bugs were reported which crashed DLITE when encountered. The first arose when users attempted to get more results from a successful search. To accomplish this, users were supposed to click on the number displayed within the result collection. This, however, did not work, and caused the current DLITE task to crash. Similarly, dropping a result collection directly onto the Interbib service caused the current task to freeze, no longer accepting input from the user. These errors resulted from errors in the communication between the front and back ends of DLITE, in the former causing an uncaught exception to be thrown, and in the latter causing the DLITE thread to sit waiting for a additional, non-existent, information to be sent from the collection. Both of these errors were fixed, and did not reappear in subsequent user tests.

Design Flaws

While over all, the test users liked using DLITE, and found it to be both a novel and powerful way of accessing and manipulating information, there were a number of places in which certain components of the system left users confused. The majority of these could be termed "learning errors" in which users initially misunderstood some aspect of DLITE's design, or the implementation of a certain component. In the case of these "learning errors" once users learned how their expectations differed from the system's design, the errors no longer surfaced. There were very few actual "use errors" in which some aspect of the DLITE design made it consistently difficult to use, regardless of whether the user was aware of the problem or not.

The first of "learning errors" arose from queries and the query constructor itself. Many users did not initially grasp the fact that the red "Q" which was created upon clicking the "create query" button was in fact the instantiation of their query. Instead, these users thought that either the circular target area of the query constructor or the entire constructor including the "subject", "author", and "title" fields were in fact the queries. This caused them to attempt dragging the query constructor and dropping it on services. Similarly, there was some confusion as to the nature of another object, the user-defined collection. It was not apparent to users what the use of such an object was, especially as it could contain both documents and queries. While users proceeded to ignore the collection problem, the nature of the query was a more fundamental misunderstanding, and had to be overcome if users were to successfully use the system.

In addition to confusion based on nature of specific components, there were problems that stemmed from the implementation of interface components and system actions. The first of these was the result of dragging an object by the graphic in its upper left corner. Users tended to view this graphic as a handle from which they could drag objects around on the screen. According to the implementation clicking and dragging from that graphic produced a copy of the object, an unexpected reaction to most users.

A number of the problems experienced by the test users arose from a lack of clear feedback about their actions. Users did not notice the outputs of some services, resulting in their repeating the action numerous times until they noticed a growing pile of documents or collections at the output point. These then had to be deleted which caused some frustration. Similar to the result feedback, some of the other affordances built into the system in order to indicate functionality were also missed. The graphics within the input fields of many services which indicated the acceptable types of input were consistently overlooked resulting in users dropping documents onto input fields which accepted only collections. Also, it was not apparent to users that clicking on the numbers of results provided in a result collection would provide them with additional documents. As a result, many thought that DLITE was only able to provide the first twenty or so results to any given query.

The only "use error" which came up with any consistency arose from the opacity of the objects within the workspace. Users noted it was difficult to drag full-sized objects onto others as the dragged objects completely obscured the target area. This was most commonly noted in reference to dragging a collection onto the trash, in which users often took three or four tries to successfully drop a collection icon onto the much-smaller trashcan icon. The opacity of objects also became problematic when users layered the objects in their workspace. This was sometimes done accidentally, for example placing a collection on top of an Interbib result, and caused users to think they had deleted the obscured interface element.

Post-Test Impressions of DLITE:

In the post-test discussions, all but one user reported they liked the DLITE interface, and would like to use it again once some revisions had been made to alleviate the problems noted above. According to users, once they understood the system and with the exception of the opacity problem, they found it fairly intuitive and easy to use. They found the workspace of DLITE to be a useful place to hold and keep track of their documents. In addition, they found the drag-and-drop manipulation of the graphical instantiations of documents to be quite easy to use, as it mimicked the manipulation of non-electronic documents.

Conclusions:

The results of the post-test interview raise a question as to what gave rise to the problems noted in the results section. Through their actions in the test and responses in the interview, users indicated that the errors and problems most commonly experienced resulted from a failure to understand the model upon which DLITE was based. Once users had grasped that model, there were comparatively few errors. This indicates that at the time of the user tests DLITE had a high learning curve, but once users passed beyond the initial learning stage, the system was relatively straightforward to use.

Based solely on users' actions in the tests, it is unclear whether the steep learning curve arose from problems with implementation or an unintuitive conceptual model, which took a long time for users to internalize. The comments given by users in the post-test interview, however, indicated that they found the interface very intuitive. Users felt that once they had figured out what the components actually were and how they functioned, DLITE was quite easy to use to accomplish the tasks they had for it. This, combined with reactions like "Oh, I see", and "That makes more sense" indicated that DLITE miscommunicated its functionality and conceptual model it expected its users to have. These comments and the observations of the actual user tests indicate that the errors and problems experienced by most users arose from shortcomings of the actual implementation of the system, not its underlying model.

The user tests further indicated sources of a number of these problems, which were as follows. The first was a misunderstanding as to what graphically represented a query constructor. The second was the failure of feedback mechanisms to accurately convey the completion of a job, and the location of its output. Third, failure of DLITE's built in affordances to indicate to users the appropriate input and output types to expect from a service. Finally, fourth, the opacity of elements in the interface especially in the context of dragging objects onto each other.

This indicates that some redesign of DLITE's interface in order to clarify these points of confusion would serve to make the interface as a whole much clearer, and reduce the steepness of the learning curve significantly.

Appendix 1:
User Test Field Notes

Test 1:

Problems:

<u>Previously configured workspaces</u>

When starting with predefined user, DLITE loaded up, but nothing typed in the query constructor was displayed & nothing was draggable. The "create query" button graphic remained clickable.
Occasionally, when restarting, crashes immediately on finding running pod.

<u>Severe Lag when new user created</u>

Created a new user, (or removed task from an old one)
Creating queries worked with severe lag, (on order of minutes)
Queries that were created were often created in wrong place & unmovable or disappeared after brief display

<u>Deadlock</u>

Query, dropped on Alta Vista generated result collection, and started filling it in but DLITE locked and would not finish displaying all entries
Ran at least 10 attempts in different accounts, all with same result
At point of deadlock, nothing is clickable apart from menus

<u>Display</u>

Those entries it did display did not display smoothly, instead it skipped clumps, leaving random documents scattered about, not even circle

<u>Menus</u>

Did not refresh correctly, resulted in ever-growing list of repeated menu items.
Seemed not to append menu with itself each time, as opposed to overwriting.

<u>Query constructor</u>

Could not get Easy Query Constructor.
Was only able to call up single field query

<u>Dragging</u>

Tried to drag item, but on start of drag, item zipped off screen
Due to slow refresh, was very difficult to drag items: Manipulable objects feeling destroyed by latency problem
Difference in dragging latencies made query constructor lose feel of single object

<u>Double Clicking</u>

On double clicking on a query, query displayed in the other window, but caused Deadlock.

Tests 2 & 3

Attempted user tests: aborted due to overall failure at startup

Logged user into administrator's account

Problems:
Reload Deadlock

Loaded up workspace & froze
Nothing responsive apart from menus

Solution:

Deleted task to start fresh
I tried to guide users through some of the lag

Query Deadlock

Created query, but deadlocked after query dropped on search engine

Result:

Ended user test with little/no useful feedback.

System Test 1

Carried out in response to failures in initial user tests

Tried consistent set of base tasks on all the systems I could find

WinNT

Query object creation
- Entered query and clicked create query button - failed
- Entered query and hit return - worked

Restarted Win95
- Found Running Pod
- Started locked up
- Queries visible on workspace but not active

HP-Unix-Walrus

Created one query, worked fine
Ran Alta-Vista search and got results

Tried to run second query but constructor would not create new query object

Created new object after numerous tries - might just have been lagged
-- Lag on order of 5 minutes

Dropped on Alta-Vista

Displayed some partial results then deadlocked

Sun-Ultra-Durgon

Created Query
On creation of second query, first Q icon disappeared.
Q's created earlier & displayed in wrong place remained where they were
- Could not be moved or removed as dragging only copied

Dropped query on Alta-Vista
Partial results returned
Deadlocked before being able to view them

MacOS-7.6

Failed creation of query
Deadlocked on creation of first query
Happened numerous times, could not get further

Linux-Snapper

Query object creation
Deadlocked when entered query and clicked create query button
Worked when hit return

Dropped query on Alta-Vista
Deadlocked when Q dropped on search engine

Sun-Ultra2-Cero

Query object creation
Worked when clicked query button
Failed when hit return

Dropped on Alta-Vista
Search worked
Display of results correct

Created second query
Hitting return worked
Created numerous queries, using both methods, all worked
Alta-Vista, Folio, & Dialog searches all worked correctly

Win-95

Query Creation
Created query object fine both ways
Query object displayed in the right place

Dropped on AltaVista
Deadlocked immediately

System Test 2

Revisions made on DLITE to take care of thread deadlocking
Second set of tests, (similar to first) carried out in response to these revisions & to generate list of fixes to be made

---

<u>Win-95</u>

Entered Query and created by clicking on button
- Query created
- Appeared in top of query constructor and then dropped to correct place
- Disappeared then reappeared

Dropped on Alta-Vista
- Folder icon appeared in upper left corner of Alta-Vista
- No result set created
- Deadlocked

Started up second time with same name
- Found running pod
- Froze on startup, nothing clickable

Restarted with different name
- Query created by hitting return
- Same results as above when dropped on Alta Vista

<u>Ultra1-Durgon</u>

Created Query
Dropped on Alta-Vista
- Crashed Netscape:
  - Bus Error

Restarted with same name
- Started frozen

<u>Win-NT</u>

Created Query
Dropped on Alta-Vista
- Created result set
- Deadlocked with partial results displayed

Re-started, new name
- Created Query
- Tried to drag query to Alta-Vista but DLITE was deadlocked

<u>UNIX-Walrus</u>

Created Query
Dropped on Alta Vista
- Started displaying results
- Deadlocked with partial results displayed

List of requested fixes:

Remove all non-functional menu items
Existence of menu items that are not meant to be used draw attention away from task
Could prove confusing/frustrating as user unsure of what he/she can actually do

Drop template and debug menus
Templates were not used by most, as they were largely not relevant to walk up first time users
Debug menu only relevant to programmers

Remove all warning
Was no warning when user selected remove all

Remove services not likely to be used
Systems still under construction should not appear in DLITE, as users want to try everything they see

Query constructor should return warning on illegal queries
Querying for certain things: non-quoted numbers fails, without clarifying warning or error message

Dragging gumball should not create copy
Not intuitive, people expect drag to drag object itself as it looks like representation to drag entire thing, not just to make a copy of it

Add collection menu item
Was no way for user to add collection to task

Test 4 - RL

Note:
Follow up test upon previous failed testing

Problems:
Tried to get additional results by clicking numbers, which caused crash

Tried to drop result set directly onto Interbib, which caused system to crash

Test 5 - DS

User test run after problems with Interbib & additional results supposedly fixed

Test run remotely in Crothers Hall, over Ethernet connection on IBM ThinkPad

Notes:

- Tests run in test administrator’s account
- Despite network, ran smoothly in first trial with no more lag than test runs in Digital Libraries lab

Problems:

- After quitting and restarting as result of accidentally quitting Netscape, (meant to close window), system hung on query
- Unable to complete query after this point

Tests 6, 7, & 8 - TBr, DR, TBa

Tests run remotely in Crothers Hall, over Ethernet connection. First on Dell desktop, second 2 on a Kiwi laptop

Notes:

- In case 8, due to time spent locking up & restarting, user ran out of time & left w/out survey

Problems:

- Network latency caused erratic & confusing dragging, users didn’t know where they were dropping items.
- Tried dropping result set on Interbib & system locked
- Some confusion as to the meaning of Q’s.

Test 9 - NG

Test run in DigLib Lab on Win machine

Notes:

- Less latency problem than in Crothers, but still ran slower than user expectations
- Test run in administrator’s account

Problems:

- Dropped result collection on Interbib, caused system lock
- Tried same action repeatedly restarting each time & got frustrated
- Thought actions had not been registered, (due to lag), so repeated again and again, causing multiple actions in the end
- Ran out of time & left after final crash w/out filling out questionnaire
- Summarizer not working

Tests 10 & 11 - JM, EP

Tests attempted in Gates hall, Pup-Cluster, on Linux-based machines

Notes:

These tests were to be run concurrently

Problems:

DLITE loaded but could not generate queries on either machine

Notes:

One user had to leave, other moved upstairs into DigLib lab

Problems:

Some confusion as to meaning of Q's
Tried to drop result collection: system froze
Summarizer not working

Tests 12 - CT

Test on Win machine in DigLib Lab

Problems:

- Dragged gumball expecting to drag object, not generate a copy of it
- Unsure how to create additional queries, thought could only do one at a time
- Kept closing Netscape window, resulted in annoying popup of new window over DLITE each time
- Dragged entire constructor, (not understanding that the Q was the query)
- Dragged result set before it was done drawing, (resulted in drawing errors)
- Closed other Netscape window
- Difficulty dropping onto occluded trash can
- Liked popup info about objects
- Didn't realize how to get more results, (by clicking on number)
- Dragged single document onto Interbib
- Didn't notice or react to images of what can be dropped onto different services
- Didn't always notice output of action.
  - As a result did some multiple times
- Didn't realize that the papers coming out of bibliography were summaries
- Dropped on summarizer and appeared to lose document
  - ** Dropped document onto summarizer, and it seemed to be accepted, but then dropped off and turned green.

Tests 13 -RS

Test on Win machine in DigLib Lab

Problems:

- Dragged gumball expecting to drag object, not generate a copy of it
- Unsure how to create additional queries, thought could only do one at a time
- Kept closing Netscape window, resulted in annoying popup of new window over DLITE each time
- Dragged entire constructor as opposed to Q
- Queried for automata & complexity theory & "Q" appeared in middle of other result set: seemed to have lost lock on where constructor really was.
- Crashed folio search with "automata & complexity theory" (also tried automata and complexity theory, with same result)
- Lost garbage due to opacity of other elements
- Dragged document onto Interbib - didn't realize what could be dropped where
- Kept closing Netscape windows, resulting in new windows occluding DLITE
- Moved from DLITE pod window, resulted in restart of pod due to going back
- Dropped on summarizer and appeared to lose document
  - See note in Test 12

Tests 14 - CM

Test on DEC-UNIX machine in DigLib Lab

Notes:

- Ran quite smoothly, no major problems to speak of
- Generally understood what was going on
- Used yellow labels constantly, gumballized much & used labels to navigate

Problems:

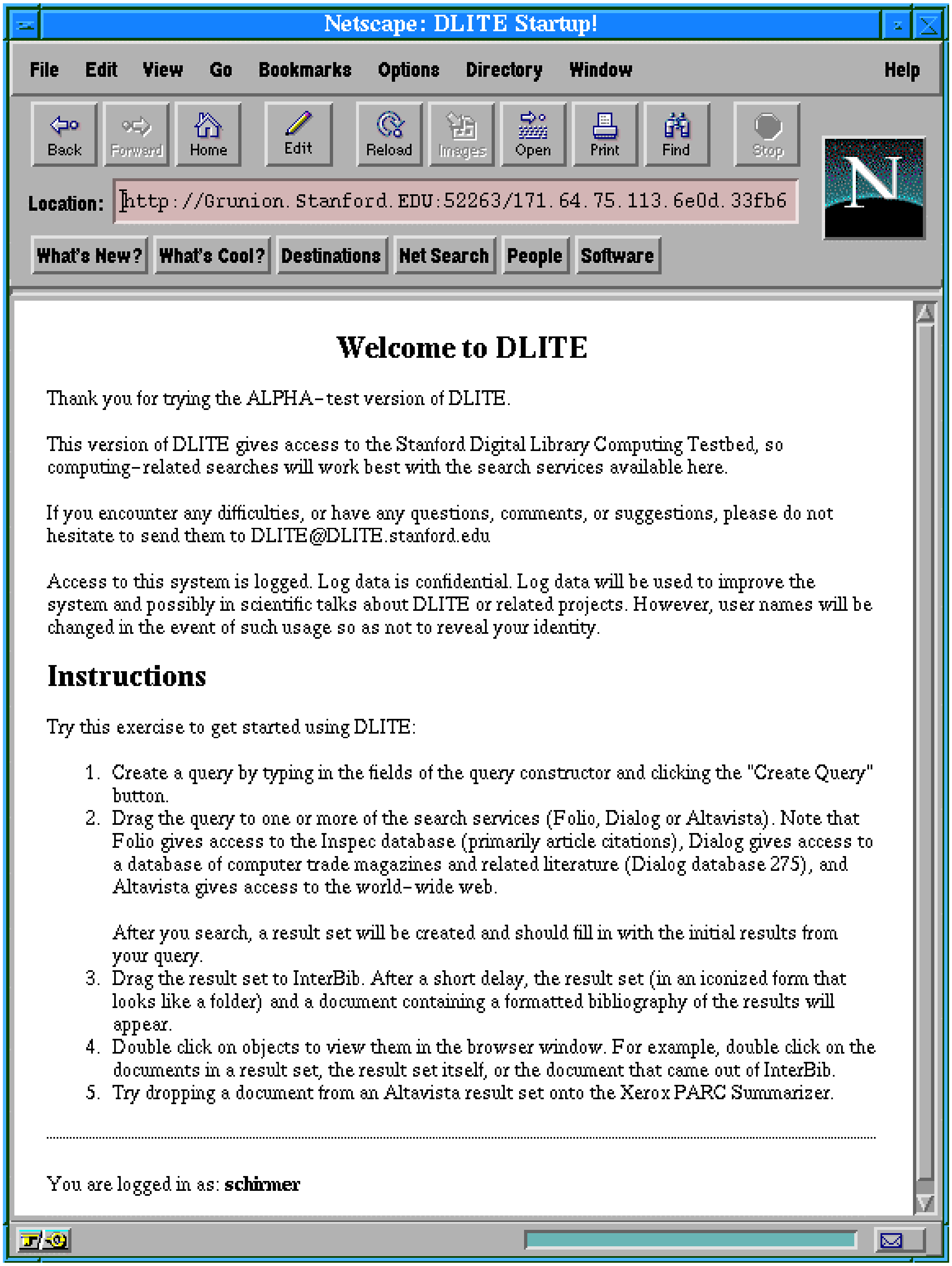
- Weak or no feedback dropping Q onto Interbib or elsewhere
- Didn't understand purpose of putting Q in collection
- Lost some items when gumballized to circles

Tests 15 - JF

Test on Win machine in DigLib Lab

- Didn't know where to drop query, tried all different things, search engines last
- Didn't understand query constructor, thought it was a query itself, and kept looking to "Results" collection to find his results
- Dropped query on summarizer and turned query green
- Complaint: Can lose data by accidentally hiding behind object
- Tried to highlight text & dragged query constructor off screen
- Didn't understand results collection purpose
- Lost Trash by double-clicking & selecting hide

Appendix 2:
Intro Screen



Welcome to DLITE

Thank you for trying the ALPHA-test version of DLITE.

This version of DLITE gives access to the Stanford Digital Library Computing Testbed, so computing-related searches will work best with the search services available here.

If you encounter any difficulties, or have any questions, comments, or suggestions, please do not hesitate to send them to DLITE@DLITE.stanford.edu

Access to this system is logged. Log data is confidential. Log data will be used to improve the system and possibly in scientific talks about DLITE or related projects. However, user names will be changed in the event of such usage so as not to reveal your identity.

Instructions

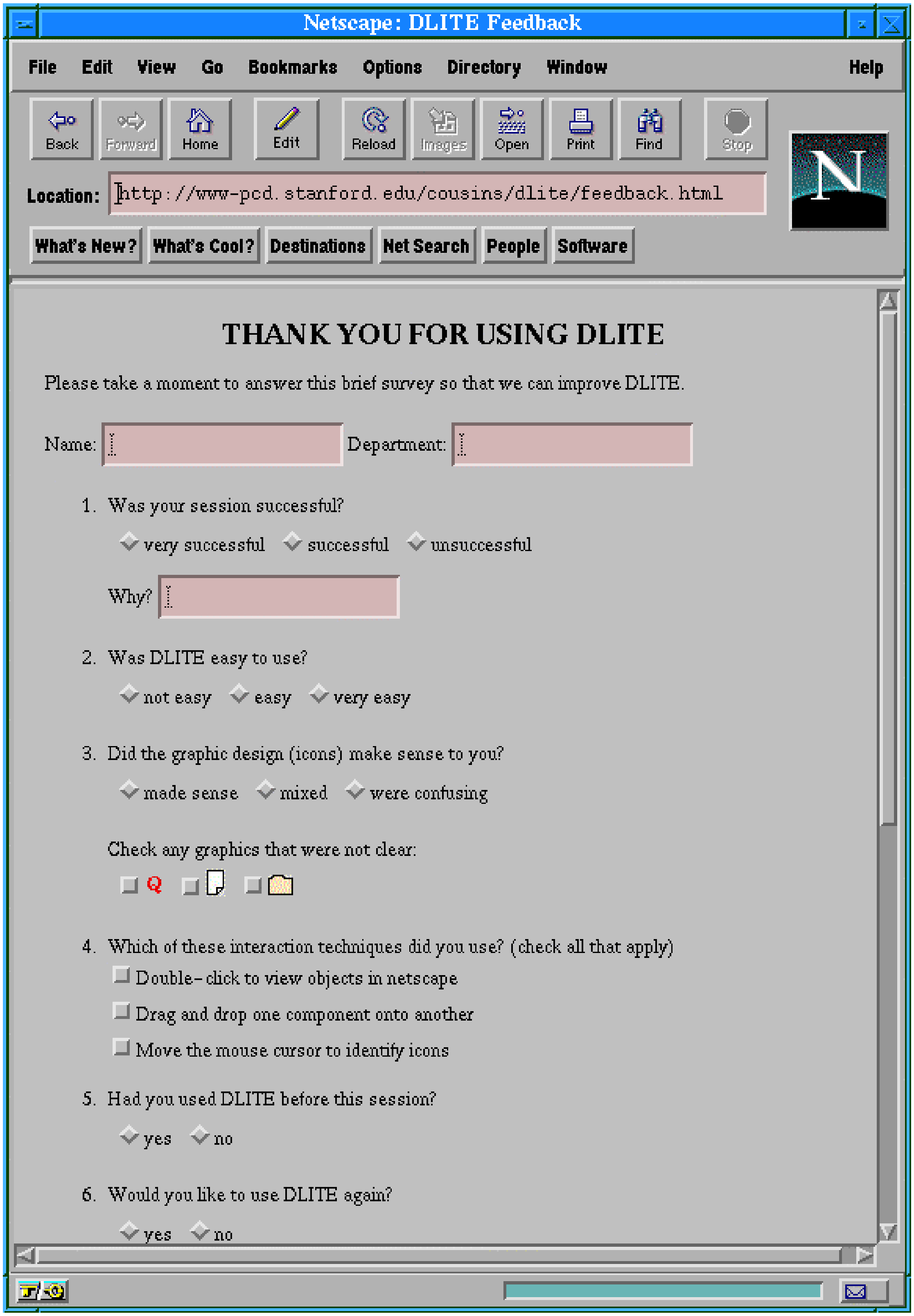
Try this exercise to get started using DLITE:

1. Create a query by typing in the fields of the query constructor and clicking the "Create Query" button.
2. Drag the query to one or more of the search services (Folio, Dialog or Altavista). Note that Folio gives access to the Inspec database (primarily article citations), Dialog gives access to a database of computer trade magazines and related literature (Dialog database 275), and Altavista gives access to the world-wide web.

   After you search, a result set will be created and should fill in with the initial results from your query.
3. Drag the result set to InterBib. After a short delay, the result set (in an iconized form that looks like a folder) and a document containing a formatted bibliography of the results will appear.
4. Double click on objects to view them in the browser window. For example, double click on the documents in a result set, the result set itself, or the document that came out of InterBib.
5. Try dropping a document from an Altavista result set onto the Xerox PARC Summarizer.

You are logged in as: **schirmer**

Appendix 3:
Final Questionnaire



THANK YOU FOR USING DLITE

Please take a moment to answer this brief survey so that we can improve DLITE.

Name: Department:

1. Was your session successful?

   very successful successful unsuccessful

   Why?

2. Was DLITE easy to use?

   not easy easy very easy

3. Did the graphic design (icons) make sense to you?

   made sense mixed were confusing

   Check any graphics that were not clear:

   Q

4. Which of these interaction techniques did you use? (check all that apply)
   - Double-click to view objects in netscape
   - Drag and drop one component onto another
   - Move the mouse cursor to identify icons

5. Had you used DLITE before this session?

   yes no

6. Would you like to use DLITE again?

   yes no