UNC
SCHOOL OF NURSING

SONIX:
School of Nursing Information Exchange

Patrick Herron
Travis Montgomery

presented 16 December 2003

SONIX Table of Contents

Executive Summary

Summary of Current System

The School of Nursing (SON) maintains its own departmental network which is linked to the campus network infrastructure. SON's network consists of 3 Novell NetWare servers, over 300 client workstations, and approximately 25 shared network printers. Each employee and doctoral student is given a SON network user id and password which allows them to access network resources. These resources include several shared folders that contain SON reference documents.

Problem Definition

The University of North Carolina (UNC)s School of Nursing is the state's flagship nursing education institution and one of the country's leading institutions of nursing research and education. 250 faculty and staff members support a student body of approximately 600 divided into undergraduate, masters, post-masters, and doctoral programs. All faculty members are assigned to one of four divisions: Academic Divisions I and II, Research Support, and Community Partnership and Practice. In addition, several units provide services to students, employees, and the healthcare community, including Student Services, Administrative Services, Continuing Education, and Information Technology Services (ITS). ITS uses Novell's NetWare to provide file and print services, provides MS Access database and application development, and provides classroom technology support.

The current system for sharing information between the various programs and departments depends largely on shared network folders. All employees are given common drive mappings to the shared folders into which shared operational documents are placed (schedules, calendars, phone lists, procedures, etc.). There are no set standards for file and directory naming or structure, and no schedule or standard procedure for updating the information. The result is a complicated file system with pseudo-descriptive names that is extremely difficult to navigate or use efficiently and which contains a lot of outdated and/or duplicated information. The shared documents need to be accessed, referred to, and used regularly by all employees. Frustration among users appears to be universal; the mere mention of "The P Drive" will interrupt any meeting with a chorus of groans and eye-rolling. Users work around the issue by calling departments and requesting information by

phone, sending important information as email attachments to everyone in the school, or often by printing and distributing hard copies.

The objective of SONIX is to provide an efficient, intuitive, and user friendly system of distributing information between the various SON departments that reduces or eliminates both outdated and duplicated information as well as the need to print hard copies of common documents. SONIX will need to run within or on the current NetWare environment and be scaleable enough to handle the information needs of the SON for the next 5 years. SONIX should also function as a portal for employees to access University and State information that is relevant to their position (HR, research, purchasing, etc). Meeting the objectives for SONIX should dramatically increase communication efficiency and reduce workload for an increasingly overworked staff and faculty.

Scope should be minimal in terms of capital outlay, as the new system must run on the existing hardware and be compatible with existing operating systems (Windows 9x, 2000, XP and NetWare). The majority, if not all, of the development and implementation should be done by existing employees of ITS. Employees of the SON must be trained appropriately to use SONIX to find and distribute information effectively. Therefore the design and development must be carefully documented. The risk involved is minimal, involving only lost person-hours of existing employees and the continuation of the status quo. The scope of user impact is quite large, however, as the user directly involves approximately 250 faculty and staff members.

A preliminary idea for SONIX is to develop a SON intranet. SONIX might act as a browser-based user-friendly interface for searching the file system and quickly accessing documents. SONIX could also include applications such as web-based departmental calendars. This would allow departments to post information easily, allow users to access SONIX through an already familiar web browser, and it could easily link to other existing systems. Any intranet must be developed to run on the open source Apache web server.

Description of Current System

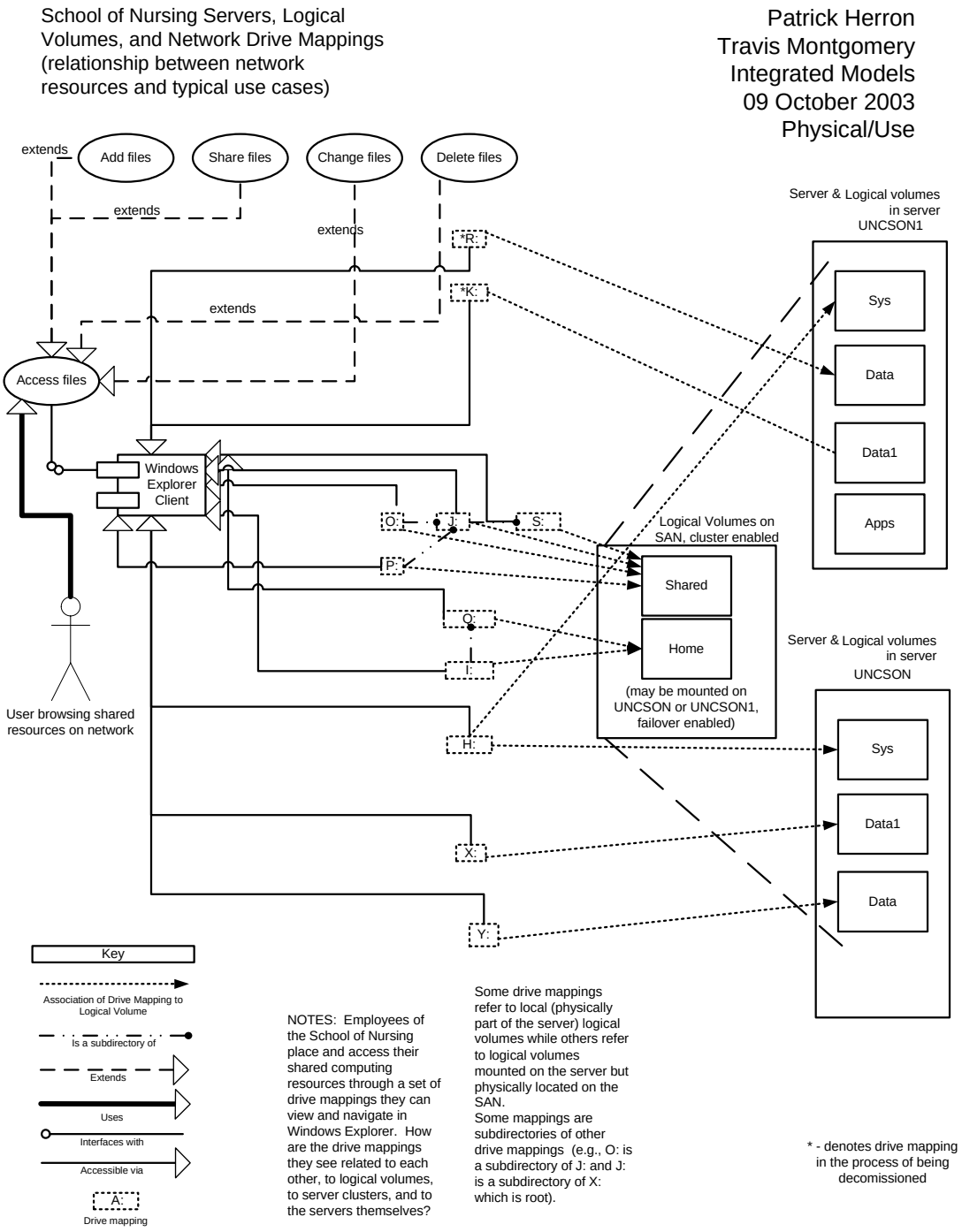
The School of Nursing uses Novell NetWare 6 and Novell Cluster Services (NCS) to provide highly reliable and secure network file and print services. The SON has two primary NetWare/NCS servers, UNCSON and UNCSON1, which are attached to a Storage Area Network (SAN). Folders which reside on the SAN can be mounted on either physical server and can be moved for load balancing purposes. In the case of a server crash, all SAN resources previously mounted on the failed server will automatically "fail-over" to the remaining server. In these cases, resources will be off-line no more than a few seconds and users generally do not notice there was ever a problem.

When a user logs-in to the network, log-in scripts provide 11 drive mappings to various network folders (see Figure 1 below). These drive mappings, each represented by a single letter, allow a user to quickly access folders on the network using Windows Explorer. The primary shared document folder drive mappings for the SON are S:, J:, O: and P:. S: is the root level shared folder on the SAN. J: is the Common folder, a sub-folder of S:. O: and P: are sub-folders of J:.

These folders are used to publish SON documents that are needed by other departments and employees. For instance, the human resource department publishes policies for determining hiring salaries for research assistants. The policy is documented then saved on the P: drive so it can be easily accessed by a project managers hiring for a research grant. Another example is the Office of Academic Affairs which publishes the class schedule for up-coming semesters, this Excel document is also saved to the P: drive. A Word document, INDEX.doc, resides in the root shared folder. This file contains a description of commonly access documents, as well as its name and location. Faculty members are referred to the document at their year opening orientation. Staff members may or may not be made aware of the document during departmental orientations.

Despite its efficiency, the system is quite complex. Many users are intimidated and confused by the number of different drive mappings and the amount data stored on the network. The system has evolved over time with no set standard of organization of file and directory naming. Names are hard to decipher, they tend to make sense only to the person who created them. There is also little organized training for new users. New faculty members attend orientation just prior to the beginning of the school year were use of the system is briefly mentioned. Staff member training is left up to the individual departments.

Figure 1: Integrated physical/use model of current system



Many users have a hard time quickly locating the information they need and are frequently unsuccessful. The develop workarounds to bypass the confusing system and still get the information they need. Take for example the project manager hiring a research assistant, instead of quickly finding the document on the P: drive, he

instead calls the faculty member in charge of the grant. She doesn't have the information so the manager then calls the human resource department which emails him the document (see Figure 2 below). Not only does this process waste time, it also introduces a second copy of the policy document which will not be updated when the policy changes. Staff members who publish information also bypass the system to guarantee that the necessary people see the document. In addition to posting next year's class schedule to the P: drive, the Academic Affairs secretary attaches the document to an email and sends it to every employee in the building (see Figure 3) despite the fact it is needed by a relatively few number of people. Like the previous example, this also introduces redundant copies of a document into the system which will quickly become outdated.

Figure 2

Sequence Model: Finding Information

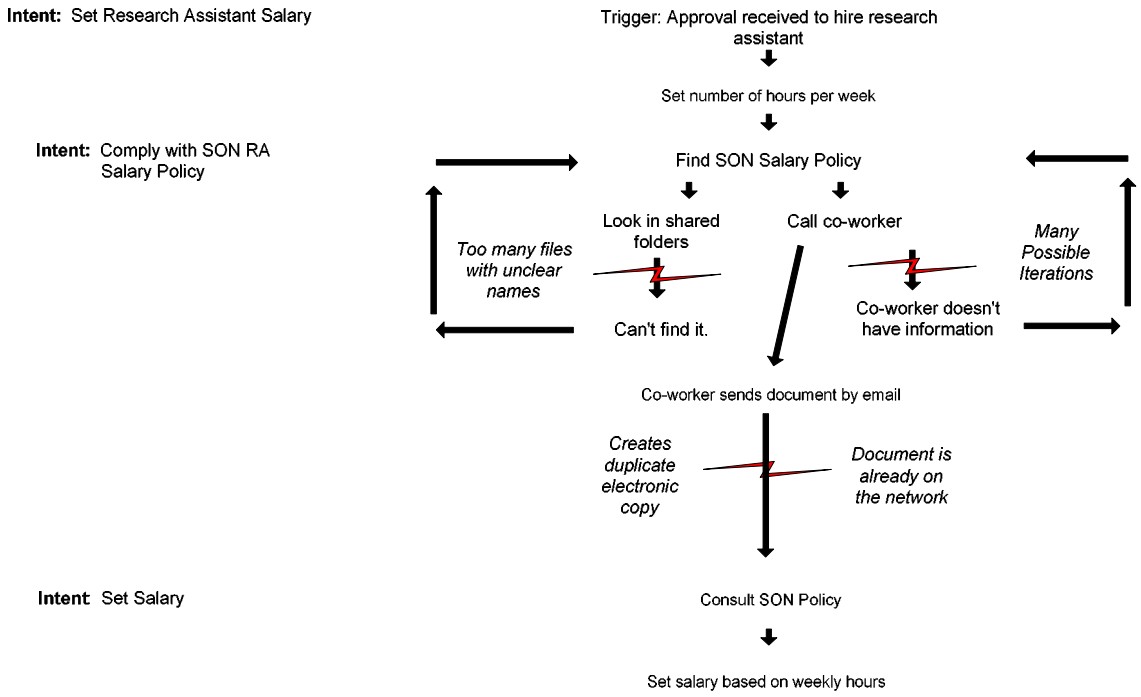
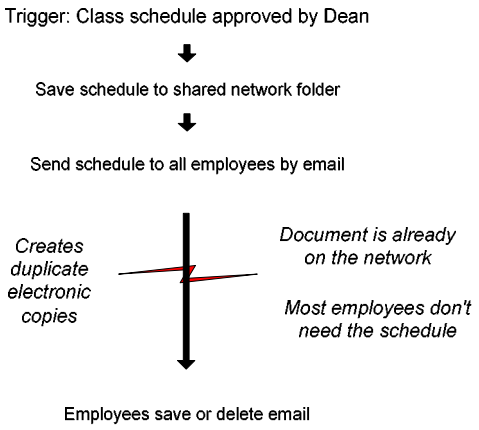


Figure 3

Sequence Model: Publishing Information

Intent: Publish next year's class schedule

Intent: Make sure everyone can find schedule



Recommendations for New System

There are three areas of change we recommend for the School of Nursing. First, the existing file system needs to be cleaned up. Currently, shared documents are all being placed in the same folders with little organization. The file names need to be made more descriptive and the files need to be logically grouped into sub folders. The existing INDEX file should be enhanced with hyperlinks and be made available in HTML format.

Summary: Clean Up File System

- Archive or delete outdated files
- Organize files into sub-directories with descriptive names
- Add hyperlinks to Index file in root folder
- Create a schedule and procedure for updating and archiving files

Second, the SON should implement Novell's Web Search Server. This software, already owned by the SON, will allow employees to search the shared folders using a familiar web browser interface. The SON can also choose to implement some optional enhancements which would give the users a customized interface which tracks most recent and most frequent searches. The search page can also form the starting point for a SON intranet and could even be pushed out to user's desktops. The default views of the search interface and search results interface can be seen in figures 4 and 5 respectively. A selected document would open within the browser (see Figure 6).

Summary: Implement NetWare Web Search Server

- Indexes both file system and internet resources
- Full text searching
- Accessed through web browser
- Integrates with existing network security

Figure 4: Novell Web Search Server Default User Search Interface

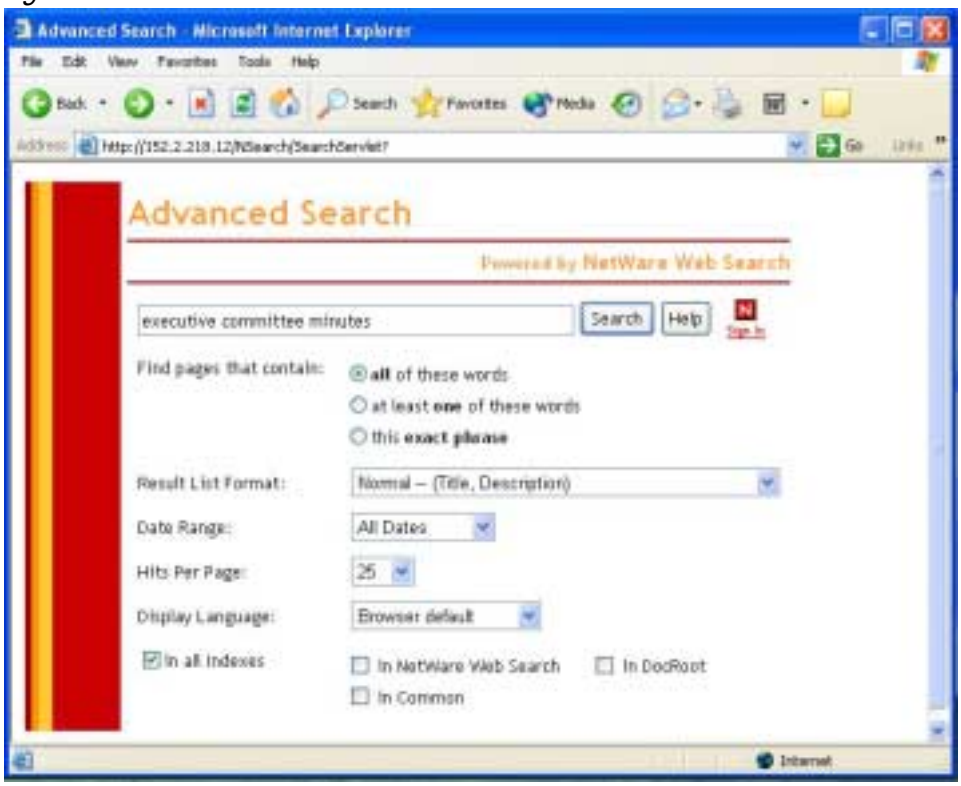


Figure 5: Novell Web Search Server Default User Results Interface

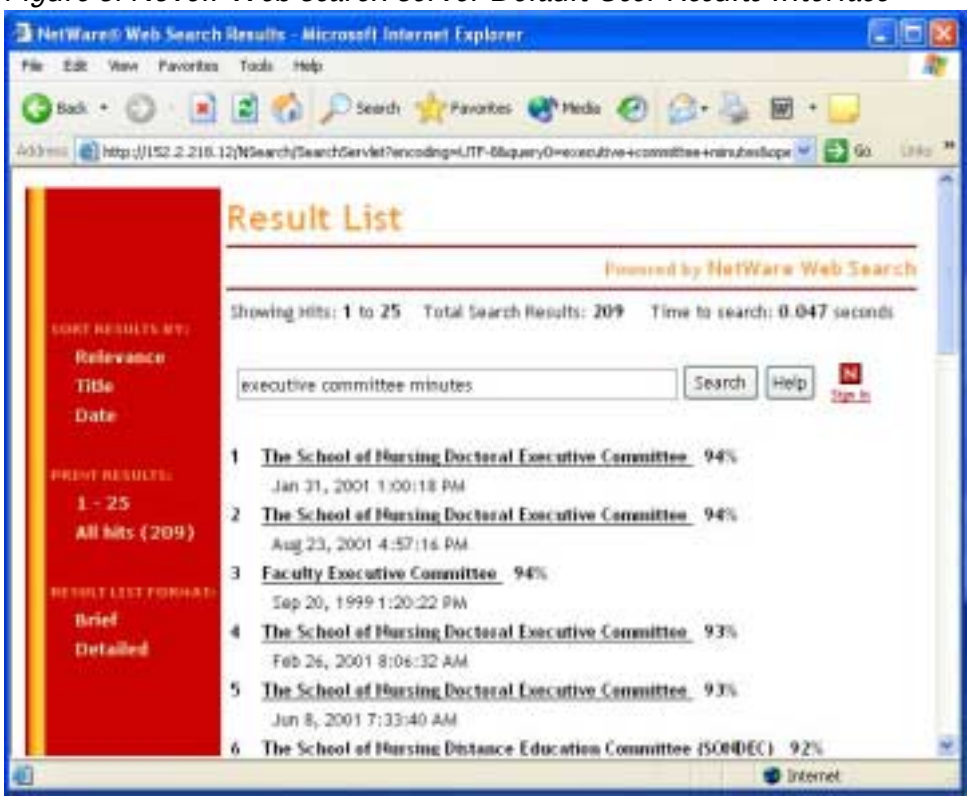
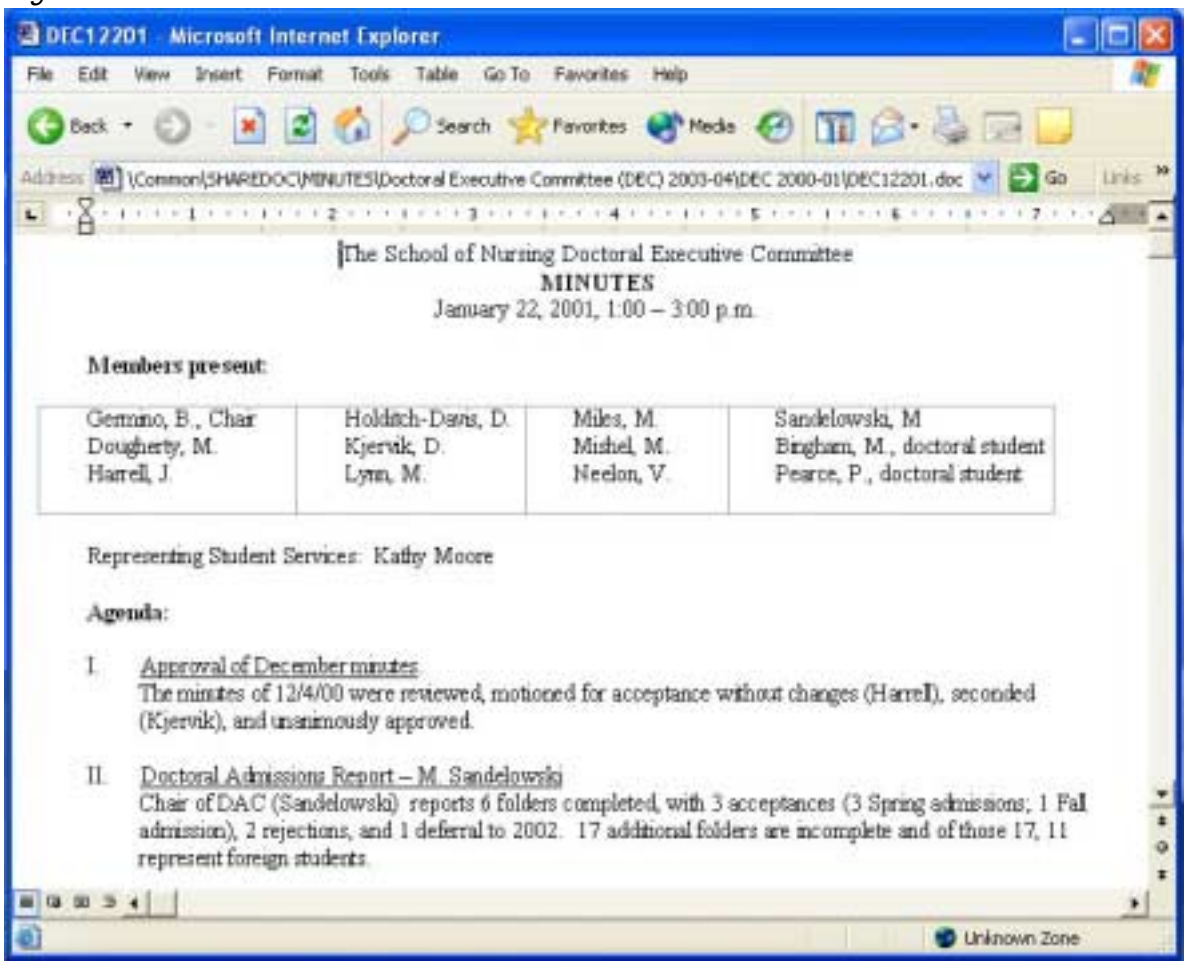


Figure 6: Product of Results Selection



Finally, SON must generate and implement documentation and training for the new search system. In order to facilitate user acceptance of the new system, the faculty and staff must be made aware of and be trained to use the new system. Announcements could be appropriately made using the SON mailing lists and ITS Special Topics Seminars should be held for initial user training. Documentation, in HTML format, should also be created and made accessible from desktop and interface links.

Summary: Documentation and Training

- Documentation in HTML accessible from link pushed out to user desktops or intranet
- ITS Special Topics Seminar
- Support through SON Helpdesk

SON has an option of implementing a custom design for the search interface, and an additional option of adding personalized portal functionality to that custom interface. The custom personalized portal features offer the opportunity to save additional productivity across SON by further reducing search times for documents regularly used by each individual. Each individual would see two personalized lists on his/her search page: a most frequently requested documents list (as a list of links to 5 documents) and a most recently requested list (another list of links to 5 documents. Both lists are for that specific user.

The optional addition can be done in one of two ways: the custom interface alone can be chosen, or the custom interface can incorporate the custom portal application. Most of the development time involves the portal application, yet most of the benefits of the custom search implementation would stem from the portal application.

Web search server runs in part within the Apache Tomcat server. Customizations of the interface and the controller components are run within the Tomcat section of the Web Search Server (see Figure 8).

The custom search will be a highly simplified version of the default search interface (see Figure 7). The search presented to the user will be simple—very similar to Google.com, but for the SON domain. All search terms will be ANDed, and the number of documents returned will be kept down to 10. Whether the search will be advanced or simple will be selected by the user and controlled by JSP.

The heart of the optional portal application is the parsing/integration unit (see Figure 8), which does the work of reading Novell NetWare audit logs and the index.idx file and then subsequently updating the request data db. Novell has APIs (known collectively as Nsure Audit) for reading audit logs in Java, C++, or Visual Basic to ease the development task.

Please refer to the custom interface snapshot (Figure 7) for how the custom interface with the portal would appear. An application design diagram for the custom portal application is shown in Figure 8, and the entity relation diagram for the portal database is shown in Figure 9. As shown in the entity relation diagram, the design for the database is incredibly simple.

Summary: Optional Enhancements

- Simplified search box integrated into an SON intranet home page

- Customized search/intranet page could include personalized history of recently accessed documents with SON "look and feel"
- Help and advanced search features would be directly available on interface

Figure 7: Custom search interface (with portal elements)

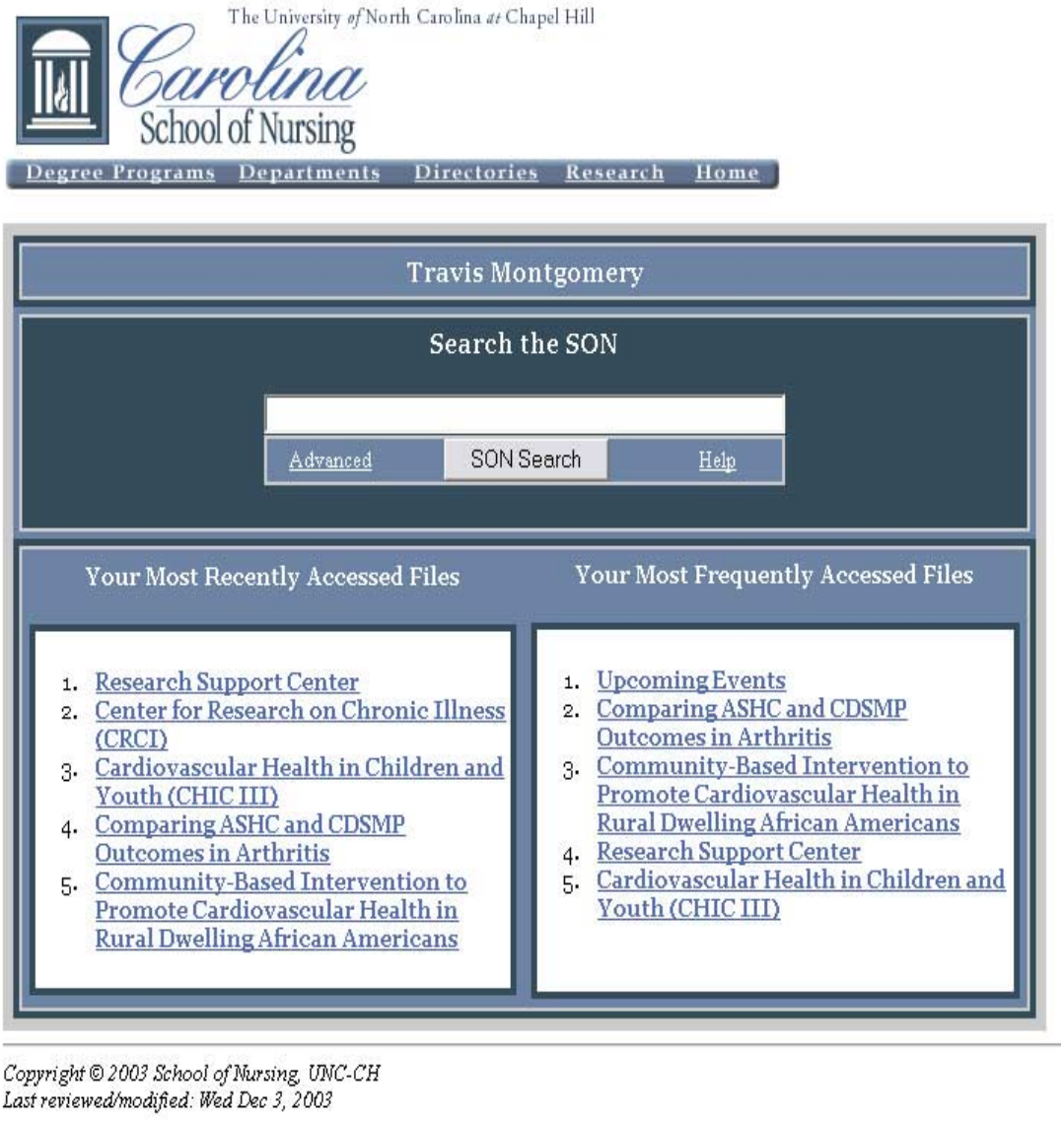


Figure 8: Optional portal application design

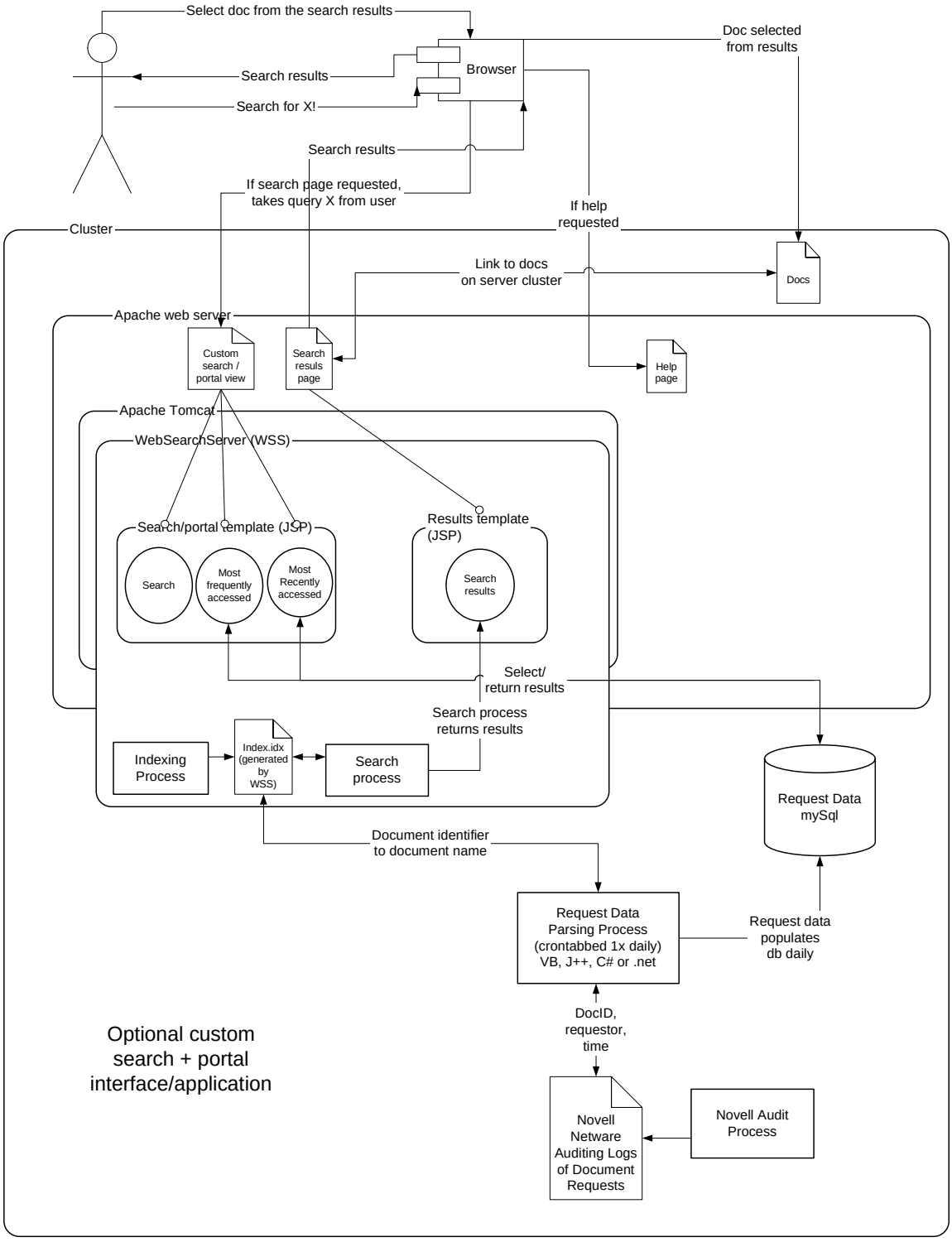
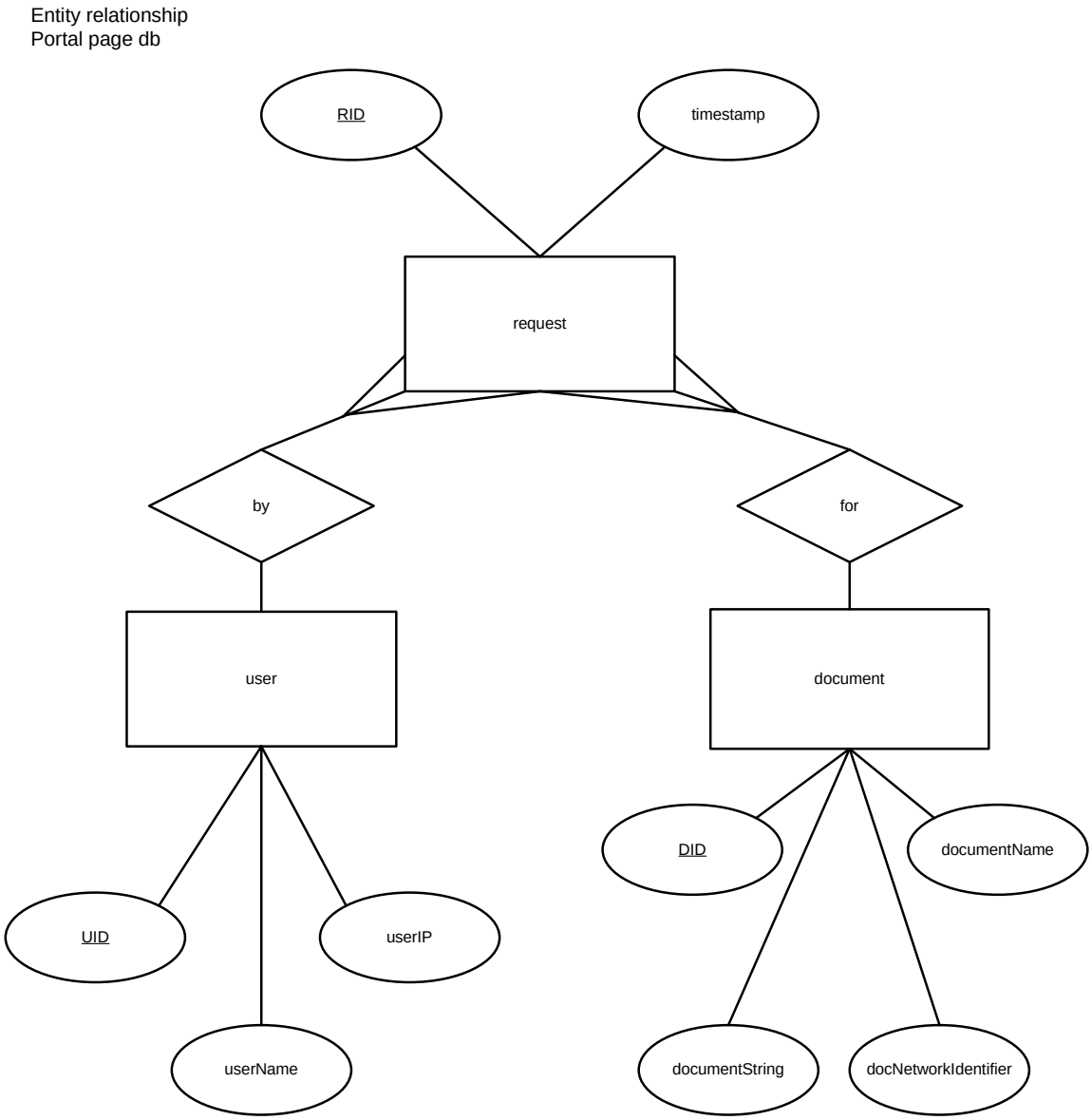


Figure 9: Entity relationship, request stats



Implementation Schedule for New System

Total time for implementation of the three main recommendations (not including the development of customized enhancements) should take approximately 1 week. The option of building a custom search interface would take an additional two days development. Adding the optional portal functionality to the custom search interface would require 8 business days in addition to the 7 business days estimated for the three recommendations plus interface. A full solution plus all options would take 3 weeks to develop, test, and implement, and can be performed using current SON staff.

Basic features implementation schedule: 5 days

1. File system rearrangement

Rearrangement of the file system is nearly complete. Any additional work can be completed as a "work in progress" and will require no further scheduling or direct allocation of resources.

2. Implementation of Novell Web Search Server

Installation of the web search server: 4 business days. The installation would be completed, tested and released by SON's IT department, spearheaded by Travis Montgomery.

Day 1: Installation of web search server
Day 2 & 3: Testing
Day 4: Release

3. Implementation of documentation and training

Day 5: Documentation assembly and delivery; automatic insertion of search link & help link to SON desktops. Additional packaging of a simple search presentation for training purposes.

Custom interface development schedule specifications: 2-10 days

4. Optional addition: custom search interface + portal additions

Development of the optional customization features will take a bit longer than the basic web search server installation. The development must come after the above three steps.

The optional addition can be done in one of two ways: the custom interface alone can be chosen, or the custom interface can be selected along with the custom portal application. Most of the development time involves the portal application, yet most of the benefits of the custom search implementation would stem from the portal application.

a. Interface development schedule: 2 days

Day 6: Simple search interface design – already completed

Day 6: Advanced search interface design – 2 hours (using simple search interface as model)

Day 6: Search results interface design – 2 hours (using search interface as model)

Interface to JSP template conversion

Days 6-7: Modification of default templates based on new interfaces – 4 hours

Day 7: Testing of search component on standard Web Search Server installation – 2 hours

b. Portal application development schedule: 8 days

Day 8: Installation of mySql

Day 8: Set up of request db

Days 8-15: Request data parsing & db population module (broken down by functionality):

Day 8: Test Novell Nsure API; Develop functionality: data mining application to read file info from index.idx file

Day 8: Develop functionality: inserting document metadata into request db

Days 9-13: Develop functionality: Mining data from Novell Netware Auditing system

Day 13: Develop functionality: Inserting request metadata into request db

Days 14-15: Testing

Benefits, Costs, Risks

A summary of benefits of the new system

- Eliminates mass distribution of email attachments and paper copies
- Reduces phone calls looking for documents
- Intuitive and up-to-date file system
- Consistent and easily accessible documentation & training
- Easy to use full text searches
- Optional enhancements provide a starting point for an SON intranet
- No-cost development for the basic solution and low-cost development for options
- Saves time and money – at least $9000 per year of lost labor just from the time employees take to use the phone to find documents on the network; more labor savings may result from the implementation of the full portal;
- Full optional development lays groundwork for enterprise-wide portal applications, including group and department level custom portal pages without requiring a purchase of the $50,000 Novell Portal software
- Full optional development also prepares SON for security analysis and network abuse monitoring due to establishment of a useful audit infrastructure and deployment of audit data analysis software

Cost

The cost of the basic implementation, already partially complete and to be done in-house, is solely the cost of allocating labor to completing the file server cleanup, search server implementation, and documentation support. Novell Web Search Server uses an existing software package that is already owned by SON. The Novell server cluster has more than adequate space to accommodate an enterprise-wide deployment of the Web Search Server.

The cost of custom implementation requires some basic HTML, graphic design, and JSP skills. Most of the HTML and almost all of the graphic design needed for all of the templates has already been created. If outsourcing is required for all portal engine development, then an additional $3-4000 in costs would be incurred.

As mentioned in the benefits section, SON stands to gain at least $9000 per year for the basic options in productivity alone. The full installation, testing and deployment of the Novell Web Search Server come at no additional cost to SON. The optional elements may carry additional costs if the current staff cannot successfully complete the development of the request data parsing module (see Figure 8).

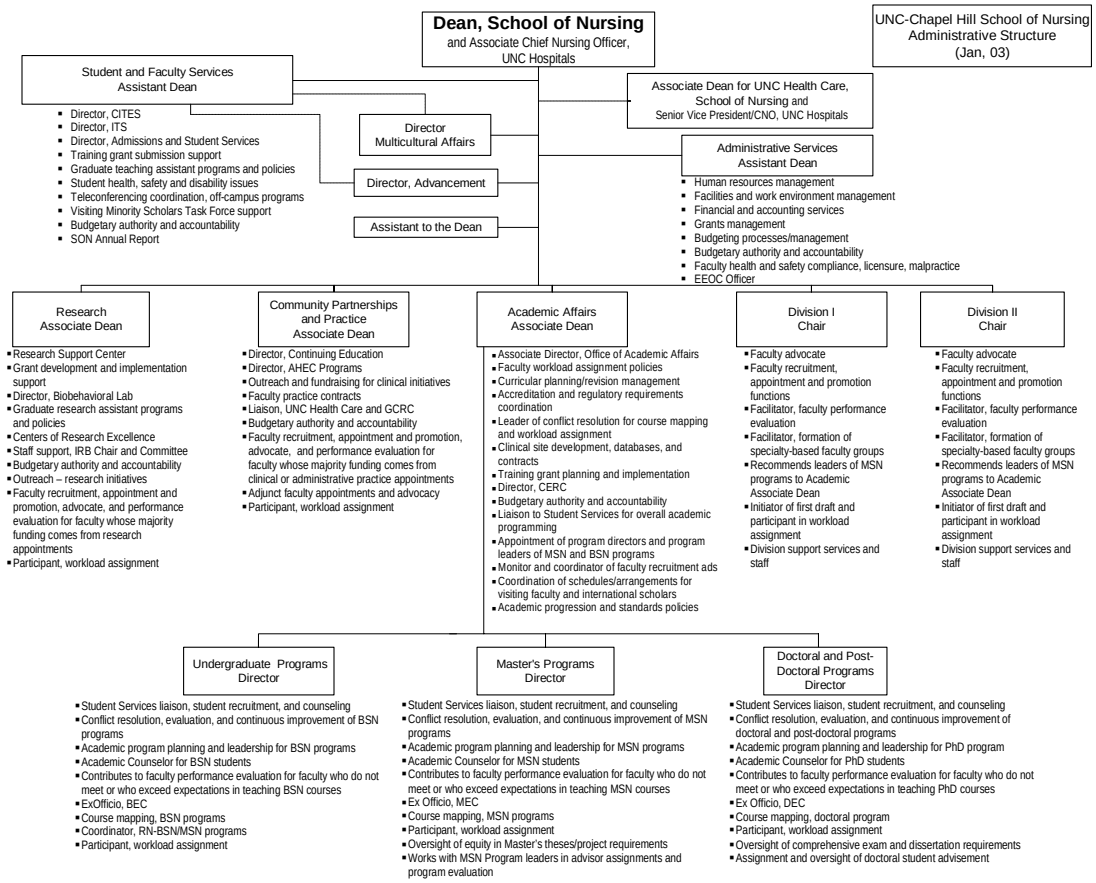
Risks

The risks of the basic option are minimal. The only significant source of risk comes from a lack of acceptance of the new search tool at SON. This risk will be mitigated in three ways: through the availability of publicly announced training opportunities provided by SON's IT department, through search training for all new employees, and through the automatic delivery of a desktop search page link to all SON desktops.

The risks of the optional enhancements are greater than the basic solution, but would significantly reduce the risk (and cost) of a future portal solution project for the SON. The existence of Novell APIs and the different language options reduce the necessity of bringing in outside help for development of the portal engine, which is the only element that might spell a development failure. Success offers the opportunity of avoiding a $50,000 cost for purchasing Novell's portal application along with reducing future portal and security development costs, while failure would merely reveal future portal development barriers for SON. In order to substantially mitigate the risk of failure of the portal development phase, it is strongly recommended that if the portal option is chosen, that a test of one simple API function be run on the Novell Audit system before a full commitment of development resources are made. As recommended in the implementation schedule, this test should be executed on day 1.

Appendix

Additional Models: Organization Chart



Additional Models: Use Case Scenarios: Finding a document

(Based on Alistair Cockburn's Main Success Use Case Scenario scheme from *Writing Effective Use Cases*. Addision Wesley, 2001.)

Find a File on the SON Network: Current System

Main Success Scenario

1. User opens Windows Explorer
2. User navigates to proper file server and directory
3. User selects the file
4. User terminates search

Extensions

1a. User goes to 4

2a. User cannot find directory
- 2a1. User browses directories on the network
  - 2a1a. Document has been renamed
    - 2a1a1. User returns to 2a1
    - 2a1a2. User goes to 2a2
    - 2a1a3. User searches, goes to 2a3
    - 2a1a4. User goes to 4
  - 2a1b. Document has been deleted
    - 2a1b1. User returns to 2a1
    - 2a1b2. User returns to 2a2
    - 2a1b3. User searches, goes to 2a3
    - 2a1b4. User goes to 4
- 2a2. User contacts coworker for correct document location
  - 2a2a. Document has been renamed, go to 2a1a.
  - 2a2b. Document has been deleted, go to 2a1b
- 2a3. User utilizes windows search tool, searches by name/substring
  - 2a3a. Search produces correct result
  - 2a3b. Search produces no results

2a3b1. Document has been renamed
2a3b1a. User modifies search query, returns to 2a3
2a3b1b. User returns to 2a1a
2a3b1c. User goes to 4
2a3b2. Document has been deleted
2a3b2a. User modifies search query, returns to 2a3
2a3b2b. User returns to 2a1a
2a3b2c. User goes to 4
2a3b3. Search query incorrect
2a3b3a. User modifies search query, returns to 2a3
2a3b3b. User returns to 2a1a
2a3b3c. User goes to 4

2a4. User goes to 4

2c. User goes to 4

4a. User initiates another search (return to 2)

Find a File on the SON Network: Proposed Basic System + Enhancements

Main Success Scenario

1. User clicks on search page icon
2. User enters search term(s) and hits enter
3. User receives results of search & selects the file from results page
4. User terminates search

Extensions

1a. User enters search page URL
1b. User hits back button after previous search to go back to search interface

2a. User selects document from list of personalized portal links

3a. User receives no search results
3b. User receives undesired search results
3b1. User initiates new search (back to 1)

3b2. User terminates search (go to 4)
3bc. User contacts coworker for document specifics

3c. User unsure of search results
3c1. User browses results list and selects a document after browsing
3c2. User initiates new search (back to 1)
3c3. User terminates search (go to 4)

4a. User initiates another search (return to 1)

Lingering Issues

While we may have misplaced our organization's cultural model (where is it!!!), it basically shows that there is a significant cultural barrier to the successful completion of the project. It is the feeling of the IT director that is someone cannot find a document with the current model, then that SON employee "isn't qualified to work here."

Fortunately, contempt for technically under-skilled or forgetful employees is not found in the remainder of SON's IT department outside of its leadership. Everyone else interviewed outside the IT department was really quite enthusiastic about the search tool (e.g., "thank god"), especially the recommended portal interface. Encountering the SON network drives directly is a stressful experience for most SON employees, and any reduction in the complexity of searching for a document is not only welcome but encouraged. The negative feelings of the IT director should in all likelihood diminished, and have done so since the initial interview in September.

Novell is not forthcoming with regard to the technical specifications of the auditing software. If the auditing feature of Novell NetWare can be harnessed, then SON stands to gain a huge leap in their ability to maintain security on their servers. User behaviors would be tracked, and such information would be available (not readily, but there) if a serious abuse of shared resources were to occur. The optional enhancements are not 100% feasible but the description and specification of the enhancements do provide the client with significant information about developing a portal system and generate excitement about taking the first step of implementing the Novell Web Search Server and cleaning up the file system.

Lessons learned

Modeling is fun.

We highly recommend the use of integrated models. We would spend more time working on the technical specs of custom portal development in a Novell NetWare file system, particularly with regard to the operation of the NetWare auditing system. The technical details are available, but Novell makes them difficult to get to, as they have an incentive to steer people towards the $50,000 portal suite they sell.

We also recommend in the future that project planning and technical specifications should be done by different people. Project plans (such as information gathering plans) often deter technical employees from working one step at a time: the scale of

such plans tends to make small projects look larger and more formidable than they are.

We still wish to learn how much technical data should be included in a client presentation. Situations vary widely in presentations: for some, technical details are *de rigueur*, while for other clients, technical details simply put them to sleep and ultimately end up killing the project. We chose to find "a happy medium" in our project: to include technical details, but only infrequently, and solely for the more complex aspects of a solution.

Most importantly, we believe the project built our confidence about the non-technical, user-friendly aspects of the project. It is so easy for tech folks like us to focus on the specs, but we know now that we can handle the more "human" side of systems analysis.