Online Community: Knowledge Management

by

Yevgeniya Zemlyakova

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2000

Author ....
Department of Electrical Engineering and Computer Science
May 17, 2000

Certified by... ....
Harold Abelson
Professor of Computer Science and Engineering, Department of
Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by..... .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Acknowledgments

I would like to thank my teammates -- my sister, Lyudmila Zemlyakova, and my best friend, Sofya Pogreb. Without them, this project would not have been possible. I would like to thank Hal Abelson, Class of 1922 Professor of Electrical Engineering and Computer Science, for putting up with the three of us. I am also very grateful to Doctor Philip Greenspun for giving us the idea for the project, encouraging us to use the ArsDigita Community System, and providing invaluable guidance.

I would like to thank the Bridge Group management for collaborating with us on the project. They have played a crucial role in helping to design and refine the system.

Finally, I would like to thank Vadim Khayms and Pubudu Wariyapola for spiritual support, supplying coffee and encouragement and helping with the writeup of this paper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The Information Age has replaced the Industrial Age. Today, companies measure their wealth by a new type of asset –– knowledge. It is more valuable to have the best information and the best knowledge in the industry rather than physical or even natural resources. More and more companies use knowledge for strategic advantage. Therefore, knowledge assets must be nurtured, preserved and used to the largest extent possible by both individuals and organizations.

Knowledge that is not accumulated, maintained and presented in a way that can be easily accessed and understood cannot be used effectively. When the challenge of efficiently managing knowledge is resolved, an organization will experience short-term benefits in the form of increasing the quality of its products and services and long-term benefits acquiring the ability to use knowledge tools in forecasting, decision making and attracting more clients.

The problem of Knowledge Management is an area of active ongoing research. Technologies that are used to address the problem include but are not limited to databases, data warehousing, data mining and intranets. This project focuses on using the available technologies to address the issue of effectively creating, managing, sharing and using the intellectual assets of organizations.

1.2 The Idea

The work described in this thesis was inspired by the idea of creating a Knowledge Management Center for a small non-profit consulting organization serving the needs of multiple clients. Consulting is the profession of selling expertise. Knowledge is thus the most valuable asset of a consulting organization. It is important to be able to efficiently share knowledge generated in the course of a project with the client, since that knowledge is effectively what the client is paying for. It is also crucial to nurture the knowledge, which has been acquired, develop and improve it, making it available for reuse in future engagements.

The Bridge Group Community System was created with the specific goal of providing strategic advice to firms in the nonprofit sector. The Bridge Group wanted to create an environment to support creation, development and sharing of knowledge, but could not find a system that would satisfy its requirements. The company was looking for a system that supports and encourages content authoring by all users rather than a designated set of administrators. The system would have to accommodate sharing of knowledge among as well as within organizations. It would have to be extensible but at the same time not require a dedicated group of technically-savvy administrators. It should have a community look and feel and have the potential of becoming a valuable resource to organizations in the non-profit industry, thus providing additional value to Bridge Group's current clients and helping cultivate new business opportunities.

This paper describes what has been accomplished as the result of the project. An innovative system for knowledge management has been designed, implemented, tested and evaluated against state-of-the-art knowledge management applications. The system attempts to satisfy the requirements listed above. It proves that a knowledge management system can be highly extensible -- in fact, knowledge can represent anything from sporadically created meeting notes that are constantly evolving, to a rigidly structured New York Times article. At the same time, the system requires very little maintenance work on the part of the site administrator, since distributes administrative duties among the user groups, which, in turn, are managed by the users themselves.

1.3 Overview

The Bridge Group Community system was designed and implemented as part of this project. The system has been used by the Bridge Group employees for over a few weeks, and they are preparing to use it on their client engagements (the results of this experiment are discussed in later chapters). The Community system aims to serve as a tool for collaboration between the Bridge Group and its Client Organizations.

The system will benefit the Bridge Group Clients in several ways:

- will provide a tool for sharing knowledge with the Bridge Group consultants, facilitating everyday tasks such as exchanging meeting notes or collaborating on a write-up of a funding proposal.
- will enable the Client Organization Employees to use and contribute to a collection of continuously evolving content relevant to the nonprofit sector.
- will provide a mechanism for Bridge Group Client Organizations to communicate with one another, potentially generating more useful content and suggesting project ideas.
- will facilitate management and sharing of knowledge within the Client Organizations by providing mechanisms to author content, collaboratively improving it, and referring to it in connection with ongoing projects.

The system will benefit the Bridge Group Consultants by:

- enabling them to develop a collection of content related to the nonprofit sector that will become invaluable to existing clients and help attract new ones.
- facilitating communication with clients, thus making it easier to perform routine project tasks -- for example, a lot of time, effort, and paper are currently needed to organize a group of several Client Organization employees and a couple of Bridge Group consultants to jointly author a funding proposal; the system makes it easy to complete this task as demonstrated in the scenario section of this paper (next chapter).
- giving clients an opportunity to interact and share their experiences of working with the Bridge Group.
- allowing them to manage a large collection of information without requiring them to devote a lot of resources to administrative tasks.
- providing a way of accumulating valuable knowledge the Bridge Group develops during the course of its projects and reusing it for future work.

The Bridge Group Community is a web-based system. Putting the system online greatly simplifies communication among Client Organizations and limits software requirements for users to a web browser. This is very important given that the system targets nonprofit organizations that cannot be expected to invest in any special equipment.

The system revolves around the idea of sharing knowledge among and within groups of users. A group can represent a single person, a 3-person project team, or a 40-person nonprofit organization. An extensive authorization module allows users to share content without giving up control and compromise proprietary information.

In terms of a scientific contribution, the Bridge Group Community system accomplishes an immensely important task – the application demonstrates that the workload of administering a community can be reduced by distributing administrative tasks among the users themselves, thus taking a step in the direction of providing greater functionality without requiring a bigger time commitment from the administrators.

We are confident that the Bridge Group Community is becoming a valuable resource that will be highly appreciated by the Bridge Group Client Organizations as well as the consultants themselves.

1.4 Division of Work

This project is a joint effort between Lyudmila Zemlyakova, Yevgeniya Zemlyakova and Sofya Pogreb. Implementation of the system was divided roughly as follows:

- Lyudmila: User Interface, integration of ACS Modules into the system, design and implementation of the Survey module, Comments, Object Ranking, and Subject classification
- Yevgeniya: representation and administration of Knowledge Types, Knowledge Fields and Knowledge Objects
- Sofya: security and authorization, Object Linking, Bookmarks, Preferred/Non-Preferred Knowledge Types, Know-and-Share Search

This paper was also written jointly, except for the Implementation section, where each member of the team described the parts of the system she was primarily responsible for.

1.5 The Paper

This paper adheres to the following format:

- **Chapter 2: Scenarios** This chapter presents several scenarios describing typical user sessions of Bridge Group Clients and Bridge Group Employees. The descriptions include snapshots of the site to convey the look and feel of the Community.

- **Chapter 3: Related Work** Knowledge Management is a very active research field. A lot of academic work has been done on the topic of Knowledge Management Systems and many systems have been implemented. A short overview of the current research is presented and several state-of-the-art systems are discussed.

- **Chapter 4: Goals** There are a lot of questions to be answered and a lot of problems to be solved in the area of Knowledge Management. Thus, it is important to clearly define this project's goals and ambitions. This chapter outlines the goals that the Bridge Group Community System was intended to achieve.

- **Chapter 5: System Design** This chapter presents an overview of the system's design, explaining the motivation behind the features and reasoning behind interesting design decisions.

- **Chapter 6: Implementation** A discussion of the implementation of the fundamental modules of the system is presented, with explanations of the choices made along the way.

- **Chapter 7: Experiment** This chapter presents the results of the two-weeks experiment conducted by several Bridge Group employees while using the system. It summarizes their feedback, changes that subsequently had to be made, and the reasons behind them.

- **Chapter 8: Competitive Analysis and Evaluation of Accomplishments** First, a feature-by-feature comparison of the Bridge Group Community System against four current state-of-the-art Knowledge Management Environments, discussed in the related work section of this paper, is conducted. It is followed by an assessment of how well the systems that were analyzed match Bridge Group's requirements and satisfy the unique constraints the company and its clients are facing.

- **Chapter 9: Conclusion** The paper is concluded by a brief discussion of possible future work and a quick description of what has been achieved and learned as a consequence of the project.

An effort has been made to make this paper enjoyable to read while providing a clear discussion of the goals of the project and what has been done to accomplish them. The authors hope this paper will help interest the reader in the fascinating world of Knowledge Management.

Chapter 2

Scenarios

This section will present five scenarios to reflect the experience of different users of the Know-and-Share system. Besides the Know-and-Share module, the Bridge Group Community Site offers a number of tools to support and enhance the clients' experience. These tools will be discussed in more detail in the section on the design and architecture of the site. The following scenarios, however, will concentrate on illustrating the functionality of the Know-and-Share module since it is the part of the site being used to support the claims made in this thesis.

1. FaykManager –– a senior employee of the FaykFirm client organization. FaykManager is the first FaykFirm employee to start using Know-and-Share. He will manage the Know-and-Share account of his company and will serve as an administrator/moderator of the content produced by other employees of Know-and-Share. We will walk through FaykManager's first time using the system.

2. BridgeGroupEmployee –– a member of the Bridge Group staff. BridgeGroupEmployee will be an active user of Know-and-Share and will routinely contribute to content creation, management and general facilitation of the clients' experience. The scenario will illustrate a BridgeGroupEmployee's typical Know-and-Share session, including some administrative tasks.

3. FaykJuniorEmployee –– an employee of FaykFirm who works with the Bridge Group on some of her projects. She does not have any administrative responsibilities for the FaykFirm group. We will illustrate FaykJuniorEmployee's first experience with Know-and-Share.

4. FaykFrequentUser –– another employee of FaykFirm who has been using Know-and-Share for quite some time. We present FaykFrequentUser's typical session. We will also demonstrate how he can add a new knowledge type, create a new object of this type and share them with other users through assigning permissions.

5. KGBFrequentUser –– an employee of KGB Nonprofit Organization who is a client of the Bridge Group. KGB Nonprofit Organization is a client of FaykFirm as well, and thus, KGBFrequentUser would like to share some information with the FaykFirm as well as with the Bridge Group. We will demonstrate the Know-and-Share's ability to support inter-organizational knowledge sharing.

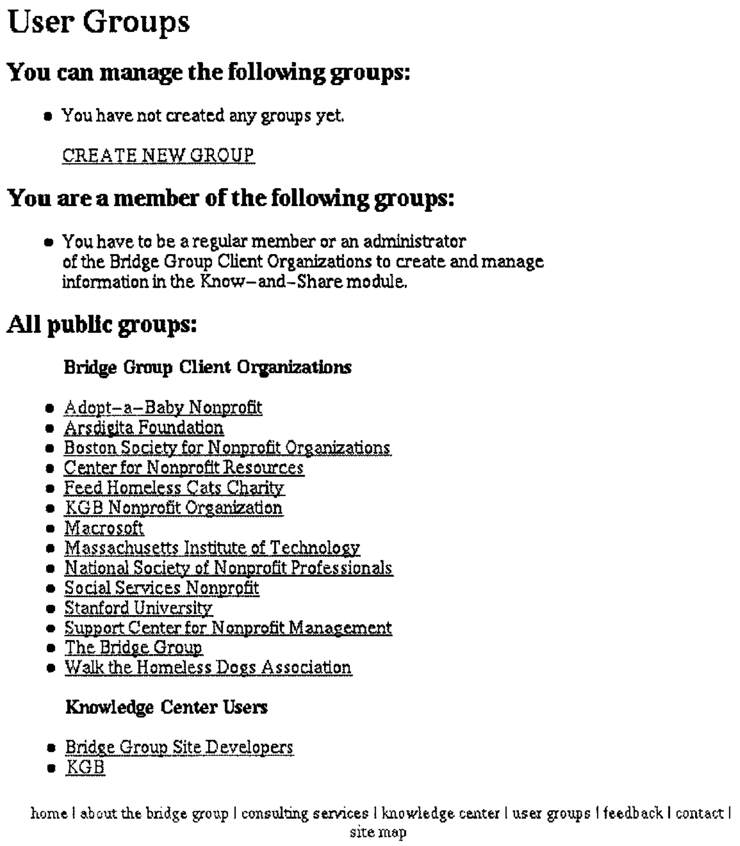
2.1 FaykManager

FaykManager opens http://lcsweb92.lcs.mit.edu/start. He has not used Know-and-Share before, and thus does not have a system account. The home page indicates that he has to be a registered member to use the Know-and-Share module. Therefore, he follows the "New Users Register Here" link and fills out the New User Registration form, which requires the user to input last name, first name, email address, password, and verification of password. He clicks the "Register" button and is redirected to the main page which indicates that he is now a registered member of the Bridge Group community. He is also provided with instructions on how to gain access to the Know-and-Share Center (he has to register with one of the existing Bridge Group Client Organizations or create a new organization if he is the first person from this organization registering with the site, and submit it for approval). Please refer to *Figure 2-13* at the end of this chapter for a snapshot of the site's home page.

FaykManager is most interested in the Know-and-Share module. He follows the instructions provided on the Registration Results page and clicks on the Client Groups button to create a new client group corresponding to the FaykFirm Organization.

Since the FaykFirm Client Organization Group does not exist yet, FaykManager follows the 'Create a New Group' link. He is presented with a choice of group types. He chooses to create a new group of type Bridge Group Client Organizations. He is presented with a five-field form which asks him for the organization's short name, pretty name, administrator's email, new member policy, and group spam policy. FaykManager wisely decides that joining the FaykFirm Organization should be subject to his approval and that spamming the group should require approval as well. These choices require more administrative duties, but they provide better security for the content created/maintained by this organization. FaykManager submits the form. The Organization group is created and submitted to the Bridge Group Team for approval. FaykManager can now proceed to use all features of the site. However, he can only view those objects in the Know-and-Share module that are available to 'All Website users' group, and he cannot add new information to the site since he has to wait for his organization's approval.

Note: The process of registering himself and the FaykFirm required him to fill out 2 5-field forms and go through five screens.



User Groups

You can manage the following groups:

- You have not created any groups yet.

CREATE NEW GROUP

You are a member of the following groups:

- You have to be a regular member or an administrator of the Bridge Group Client Organizations to create and manage information in the Know-and-Share module.

All public groups:

Bridge Group Client Organizations

- Adopt-a-Baby Nonprofit
- Arsdigita Foundation
- Boston Society for Nonprofit Organizations
- Center for Nonprofit Resources
- Feed Homeless Cats Charity
- KGB Nonprofit Organization
- Macrosoft
- Massachusetts Institute of Technology
- National Society of Nonprofit Professionals
- Social Services Nonprofit
- Stanford University
- Support Center for Nonprofit Management
- The Bridge Group
- Walk the Homeless Dogs Association

Knowledge Center Users

- Bridge Group Site Developers
- KGB

home | about the bridge group | consulting services | knowledge center | user groups | feedback | contact | site map

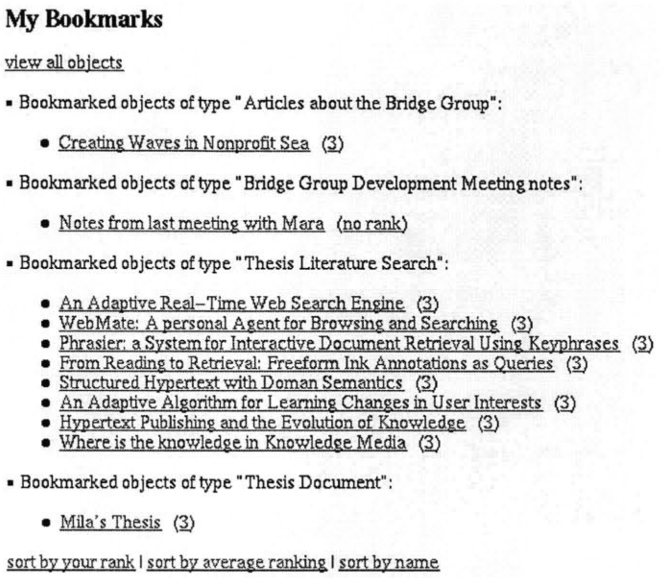
Figure 2-1: User's Personal Listing of Client Groups

2.2 BridgeGroupEmployee

BridgeGroupEmployee has been using the Bridge Group Community Site for some time and is visiting the Know-and-Share Center today to review the meeting notes from his meeting with SmallCorpManager that took place a few days ago. He also wants to post an article about the Bridge Group he read in the New York Times this morning. BridgeGroupEmployee logs in and follows the Client Groups link to check if there are any Client Organizations waiting for registration approval. He sees that FaykManager has submitted a request for creation of the FaykFirm Client Organization Group. BridgeGroupEmployee checks his records, confirms that FaykFirm is a client and FaykManager is the senior employee in charge of the Bridge Group's engagement with Manager. He approves the application and an email is immediately sent to FaykManager. This administrative task requires him to go through one screen to verify the email address submitted by this user and one click to approve him.

BridgeGroupEmployee then follows the link to the Know-and-Share Center. He notices that he can exclude the 'Url Link' object type from his list of preferred object types since he never uses the objects of this type. It will go into the excluded object types section at the bottom of the page. Please, refer to *Figure 2-14* at the end of this chapter for a snapshot of the Know-and-Share index page.

He proceeds to click on the 'My Bookmarks' link on the BridgeBrowse Bar, sorts the list by Object Rank and quickly finds the meeting notes he is looking for.



My Bookmarks

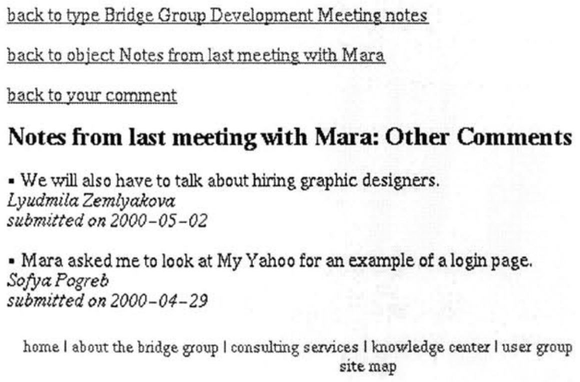
view all objects

- Bookmarked objects of type "Articles about the Bridge Group":
  - Creating Waves in Nonprofit Sea (3)
- Bookmarked objects of type "Bridge Group Development Meeting notes":
  - Notes from last meeting with Mara (no rank)
- Bookmarked objects of type "Thesis Literature Search":
  - An Adaptive Real–Time Web Search Engine (3)
  - WebMate: A personal Agent for Browsing and Searching (3)
  - Phrasier: a System for Interactive Document Retrieval Using Keyphrases (3)
  - From Reading to Retrieval: Freeform Ink Annotations as Queries (3)
  - Structured Hypertext with Doman Semantics (3)
  - An Adaptive Algorithm for Learning Changes in User Interests (3)
  - Hypertext Publishing and the Evolution of Knowledge (3)
  - Where is the knowledge in Knowledge Media (3)
- Bookmarked objects of type "Thesis Document":
  - Mila's Thesis (3)

sort by your rank | sort by average ranking | sort by name

Figure 2-2: My Bookmarks Page

BridgeGroupEmployee opens the Meeting Notes and notices that there's a comment on the Notes column made by his co-workers when they modified the notes earlier. For a full view of this page, please refer to *Figure 2-15* at the end of this chapter. He then follows the link and reviews the public comments for the meeting notes.



back to type Bridge Group Development Meeting notes

back to object Notes from last meeting with Mara

back to your comment

Notes from last meeting with Mara: Other Comments

- We will also have to talk about hiring graphic designers.
Lyudmila Zemlyakova
submitted on 2000-05-02

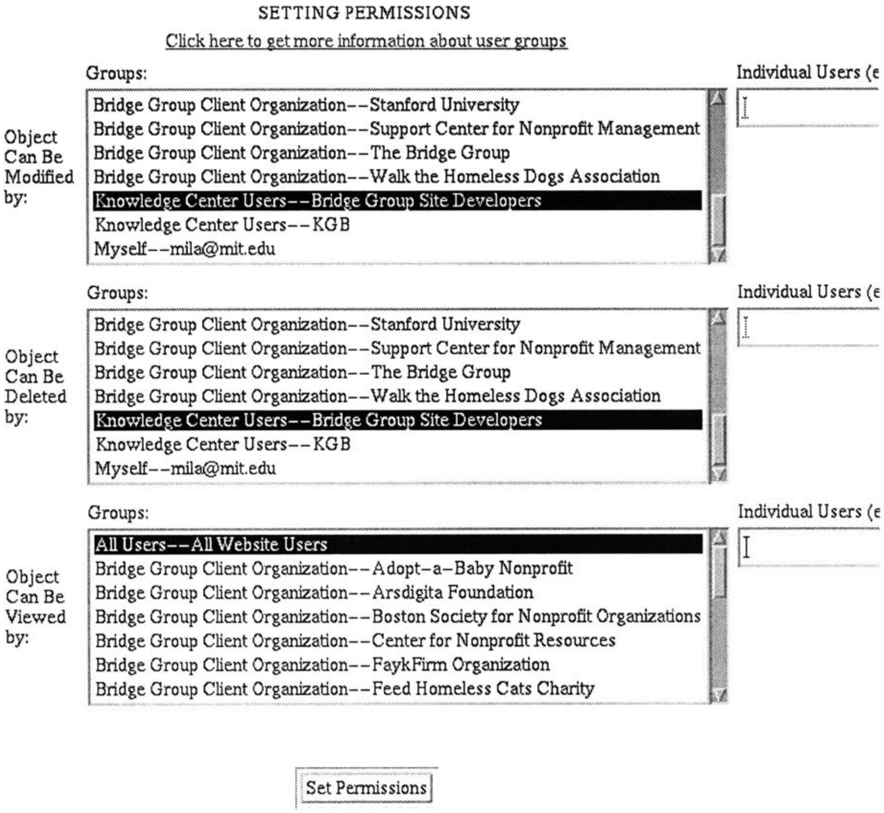
- Mara asked me to look at My Yahoo for an example of a login page.
Sofya Pogreb
submitted on 2000-04-29

home | about the bridge group | consulting services | knowledge center | user group
site map

Figure 2-3: Public Comments Page

His next task is to add an article about the Bridge Group published in the New York Times. Thus, he goes back to the listing of Knowledge Types, clicks on the 'Articles' object types and follows the 'Create a New Knowledge Object' link in order to submit a new article about the Bridge group.

He is presented with a form which asks him to fill in the fields for an article (some fields are mandatory). He types in the Article info and submits the new article. The last piece of information that he has to specify is who is authorized to modify/view this article. There are three fields to specify: who can modify, who can delete and who can view this article. He decides to make it available to All Web Site Users since he feels the article is quite well-written and entertaining and will be interesting to all users. BridgeGroupEmployee gives other Bridge Group Employees the rights to manage the object. He notices that he can also assign various permissions to different organizations, as well as to individual users (he can enter email addresses into provided form fields).



SETTING PERMISSIONS

Click here to get more information about user groups

Object Can Be Modified by:

Groups:
- Bridge Group Client Organization--Stanford University
- Bridge Group Client Organization--Support Center for Nonprofit Management
- Bridge Group Client Organization--The Bridge Group
- Bridge Group Client Organization--Walk the Homeless Dogs Association
- [✓ Knowledge Center Users--Bridge Group Site Developers]
- Knowledge Center Users--KGB
- Myself--mila@mit.edu

Individual Users (e

Object Can Be Deleted by:

Groups:
- Bridge Group Client Organization--Stanford University
- Bridge Group Client Organization--Support Center for Nonprofit Management
- Bridge Group Client Organization--The Bridge Group
- Bridge Group Client Organization--Walk the Homeless Dogs Association
- [✓ Knowledge Center Users--Bridge Group Site Developers]
- Knowledge Center Users--KGB
- Myself--mila@mit.edu

Individual Users (e

Object Can Be Viewed by:

Groups:
- [✓ All Users--All Website Users]
- Bridge Group Client Organization--Adopt-a-Baby Nonprofit
- Bridge Group Client Organization--Arsdigita Foundation
- Bridge Group Client Organization--Boston Society for Nonprofit Organizations
- Bridge Group Client Organization--Center for Nonprofit Resources
- Bridge Group Client Organization--FaykFirm Organization
- Bridge Group Client Organization--Feed Homeless Cats Charity

Individual Users (e

Figure 2-4: Setting Object Permissions Page

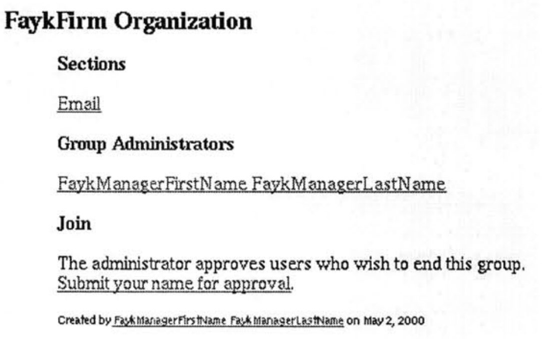
He submits the authorization form and is redirected to the 'View All Knowledge Object' page. BridgeGroupEmployee makes sure the article looks good and bookmarks it for his future use.

The process of creating a new object required filling out two forms. One form has three permission fields to fill out and the article form happened to have seven fields (five of which are optional).

2.3 FaykJuniorEmployee

FaykJuniorEmployee is working on a project with several Bridge Group consultants. Her boss, FaykManager, instructed her to visit the Bridge Group Community site and register to become a user of the FaykFirm Organization in order to use all features of the Know-and-Share Center.

FaykJuniorEmployee opens http://lcsweb92.lcs.mit.edu/start. She clicks on the 'New Clients Register Here' link, fills out and submits the registration form. Then, following the directions on the main page, she proceeds to the 'Client Groups' page and applies to become a member of the FaykFirm Client Organization Group.



FaykFirm Organization

Sections

Email

Group Administrators

FaykManagerFirstName FaykManagerLastName

Join

The administrator approves users who wish to end this group.
Submit your name for approval.

Created by FaykManagerFirstName FaykManagerLastName on May 2, 2000

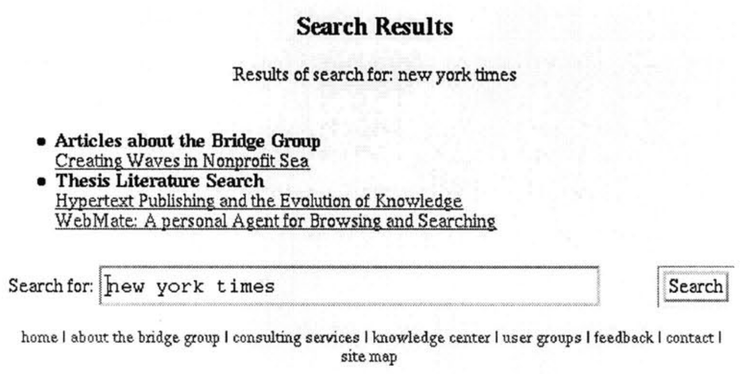
Figure 2-5: Apply to Join Organization

FaykJuniorEmployee returns to the Bridge Group Community home page, which presents an overview of features available to member of the Bridge Group Community. The BridgeBrowse bar located at the top of the page is meant to become the user's best friend –– it is present on most of the pages that are part of the Bridge Group site and helps to facilitate navigation as well as to keep the user aware of the options currently available. Below the BridgeBrowse bar is a list of tools and features available to the user. Each listing is accompanied by a short description in the 'Regular Features' section. The title page is intended to introduce the user to the site and to say "Welcome Back! Here is what's new!" every time he/she comes back. FaykManager can immediately start using all of the site's features except contributing information to the Know-and-Share center because her membership in the FaykFirm client group requires approval. Later that day, FaykJuniorEmployee receives an email informing her that her application for FaykFirm Client Organization Group membership has been approved. She can now start actively using Know-and-Share.

Note: The process of registering with the site and with a specific organization required FaykJuniorEmployee to go through 4 screens and fill out a 5-field form to register as a user.

2.4 FaykFrequentUser

As mentioned above, FaykFrequentUser has been using the system for some time and familiar with all of its features. First, FaykFrequentUser logs in and proceeds to the Know-and-Share Center. A colleague mentioned to him that there was an article about the Bridge Group in the New York Times this morning. He uses the Search tool to quickly locate the article.



Search Results

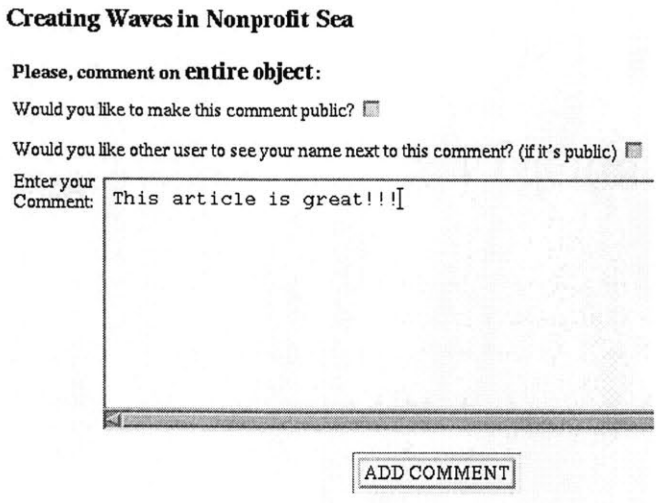
Results of search for: new york times

- **Articles about the Bridge Group**
  Creating Waves in Nonprofit Sea
- **Thesis Literature Search**
  Hypertext Publishing and the Evolution of Knowledge
  WebMate: A personal Agent for Browsing and Searching

Search for: new york times
Search
home | about the bridge group | consulting services | knowledge center | user groups | feedback | contact | site map

Figure 2-6: Search Results Page

FaykFrequentUser skims the article and ranks it as 'Liked'. Then he comments on the entire article and decides to share his thoughts with others by making his comment public.



Creating Waves in Nonprofit Sea

Please, comment on entire object:

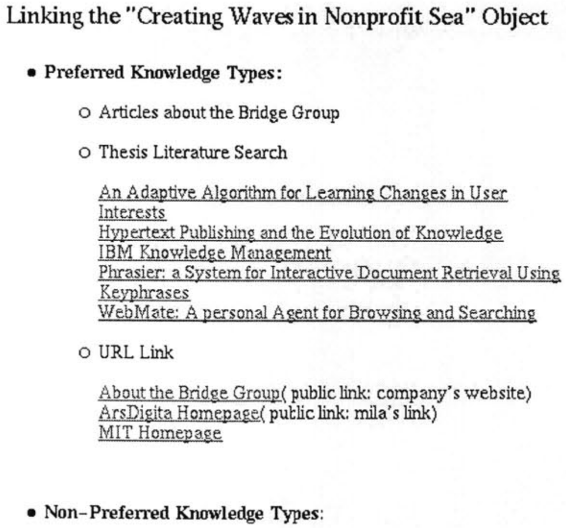
Would you like to make this comment public? □

Would you like other user to see your name next to this comment? (if it's public) □

Enter your Comment: This article is great!!!

Figure 2-7: Providing a Comment for an Entire Article

Next, he would like to link this article to other similar objects for easier navigation in the future. Thus, he follows the 'Link this Object' hyperlink and searches through other objects to find related objects to link to this article.



Linking the "Creating Waves in Nonprofit Sea" Object

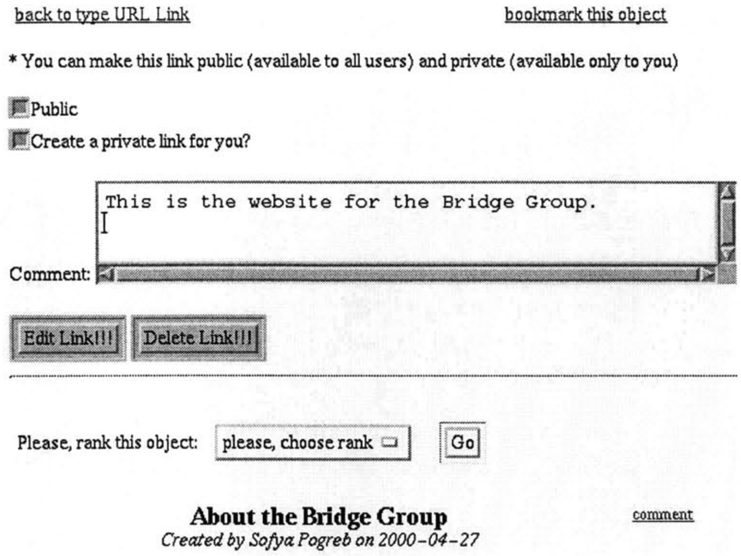
- **Preferred Knowledge Types:**
  - Articles about the Bridge Group
  - Thesis Literature Search

    An Adaptive Algorithm for Learning Changes in User Interests
    Hypertext Publishing and the Evolution of Knowledge
    IBM Knowledge Management
    Phrasier: a System for Interactive Document Retrieval Using Keyphrases
    WebMate: A personal Agent for Browsing and Searching
  - URL Link

    About the Bridge Group(public link: company's website)
    ArsDigita Homepage(public link: mila's link)
    MIT Homepage
- **Non-Preferred Knowledge Types:**

Figure 2-8: Searching To Link an Article

Then he links the article to another object he feels is related. FaykFrequentUser feels others may find the link useful, so he makes it public.



* You can make this link public (available to all users) and private (available only to you)

☑ Public

☑ Create a private link for you?

Comment: This is the website for the Bridge Group.

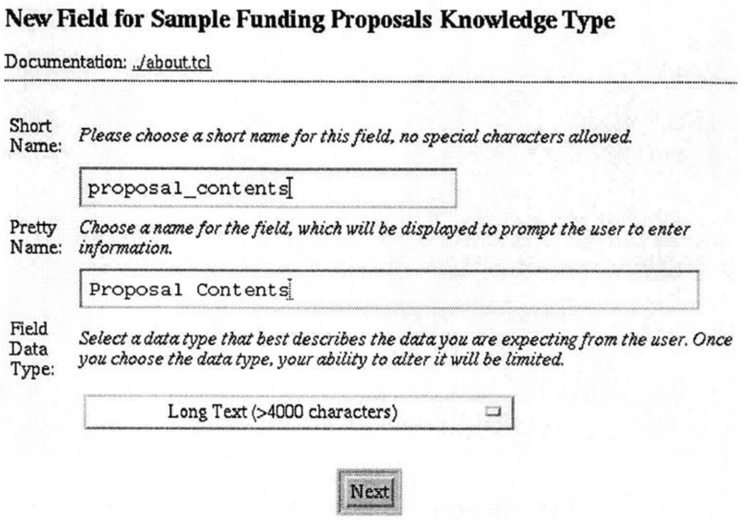
Please, rank this object: please, choose rank

About the Bridge Group

Created by Sofya Pogreb on 2000-04-27

Figure 2-9: Linking a New York Times Article to the Bridge Group Home Page

FaykFrequentUser has recently submitted a funding proposal to a government agency. He felt that having a resource with sample proposals and strategies for successfully obtaining funding would have been invaluable to him. He would like to post his proposal for others to review but the Article Knowledge Type is the closest thing available and he doesn't feel a funding proposal fits the structure of an article very well. Thus, he decides to create a new Knowledge Type –– Sample Funding Proposals. FaykFrequentUser follows the KM Administration link, chooses to create a new Knowledge Type, and is presented with a New Knowledge Type form. He fills out the form that requires him to input the short name and pretty names for the new type. He also has to distribute permissions to modify and delete this object type. He decides that the newly created objects could be modified by FaykFirm and the Bridge Group employees and deleted by FaykFirm employees. He is redirected to the page to view this type and add new fields. 'Name' and 'Overview' are provided as default fields. For each new field he has to fill out two short forms. The first form asks him to provide short name, pretty name, and the data type for this field (text, integer, date, etc).



New Field for Sample Funding Proposals Knowledge Type

Documentation: ../about.tcl

Short Name: *Please choose a short name for this field, no special characters allowed.*

proposal_contents

Pretty Name: *Choose a name for the field, which will be displayed to prompt the user to enter information.*

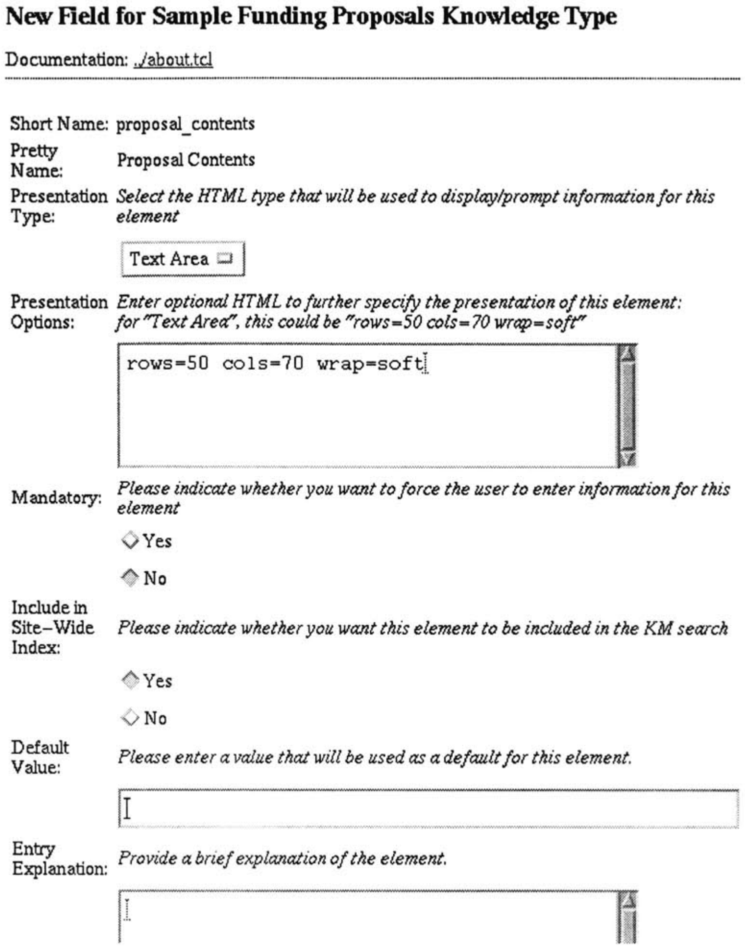
Proposal Contents

Field Data Type: *Select a data type that best describes the data you are expecting from the user. Once you choose the data type, your ability to alter it will be limited.*

Long Text (>4000 characters)

Figure 2-10: Form to Provide Basic Field Information

The other form asks him how this field should be displayed for the user. Note that all the fields on this form are optional or have already been filled out for the user.



New Field for Sample Funding Proposals Knowledge Type

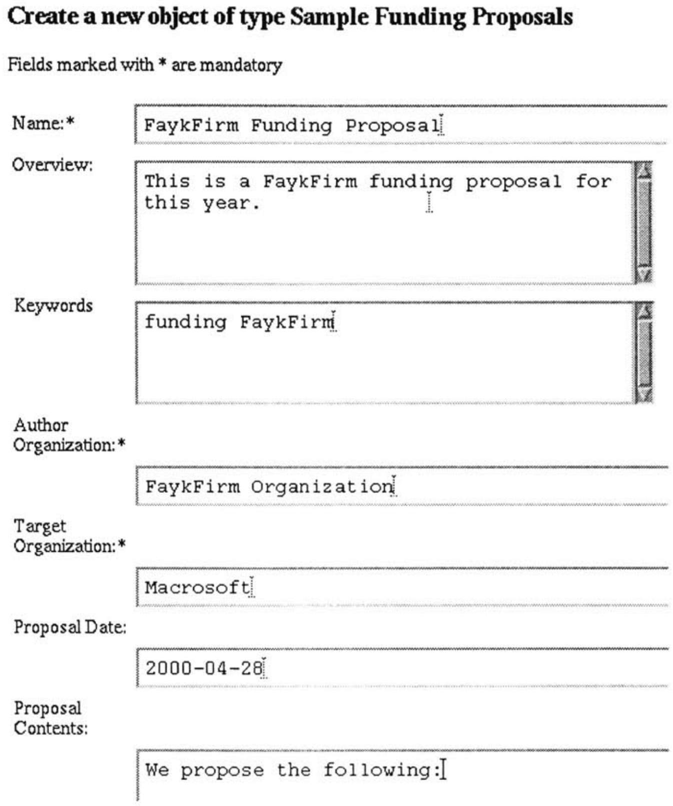
Documentation: ../about.tcl

Short Name:	proposal_contents
Pretty Name:	Proposal Contents
Presentation Type:	*Select the HTML type that will be used to display/prompt information for this element* Text Area
Presentation Options:	*Enter optional HTML to further specify the presentation of this element: for "Text Area", this could be "rows=50 cols=70 wrap=soft"* rows=50 cols=70 wrap=soft
Mandatory:	*Please indicate whether you want to force the user to enter information for this element* ○ Yes ● No
Include in Site-Wide Index:	*Please indicate whether you want this element to be included in the KM search* ● Yes ○ No
Default Value:	*Please enter a value that will be used as a default for this element.*
Entry Explanation:	*Provide a brief explanation of the element.*

Figure 2-11: Form to Provide HTML layout for the Field

FaykFrequentUser proceeds to add fields to the newly created type. He adds Author Organization, Target Organization, Proposal Date, Proposal Contents and, finally, Proposal Status. Thus, the new Knowledge Type, Sample Funding Proposals, is now ready for new content. However, in order to avoid creating a large number of redundant object types, each newly created type has to be approved by the Bridge Group administrator. Therefore, FaykFrequentUser has to wait until the administrator visits the KM admin page and clicks on the 'approve' link next to this object type. When this type has been approved, an email notification is sent to FaykFrequentUser that he can now create objects of this type. Please refer to *Figure 2-16* to view the administration page for this type.

Some time later FaykFrequentUser returns to the listing of types and chooses to create a new Object of type Sample Funding Proposals. He enters his proposal, inputting the name of the funding source and the word "funding" as keywords, and designates appropriate permissions (a form with three fields: who can view, modify, and delete this object). The required fields for this object type are marked with (*). Finally, he submits the article.



Create a new object of type Sample Funding Proposals

Fields marked with * are mandatory

Name:*	FaykFirm Funding Proposal
Overview:	This is a FaykFirm funding proposal for this year.
Keywords	funding FaykFirm
Author Organization:*	FaykFirm Organization
Target Organization:*	Macrosoft
Proposal Date:	2000-04-28
Proposal Contents:	We propose the following:

Figure 2-12: Form to Create a New Proposal

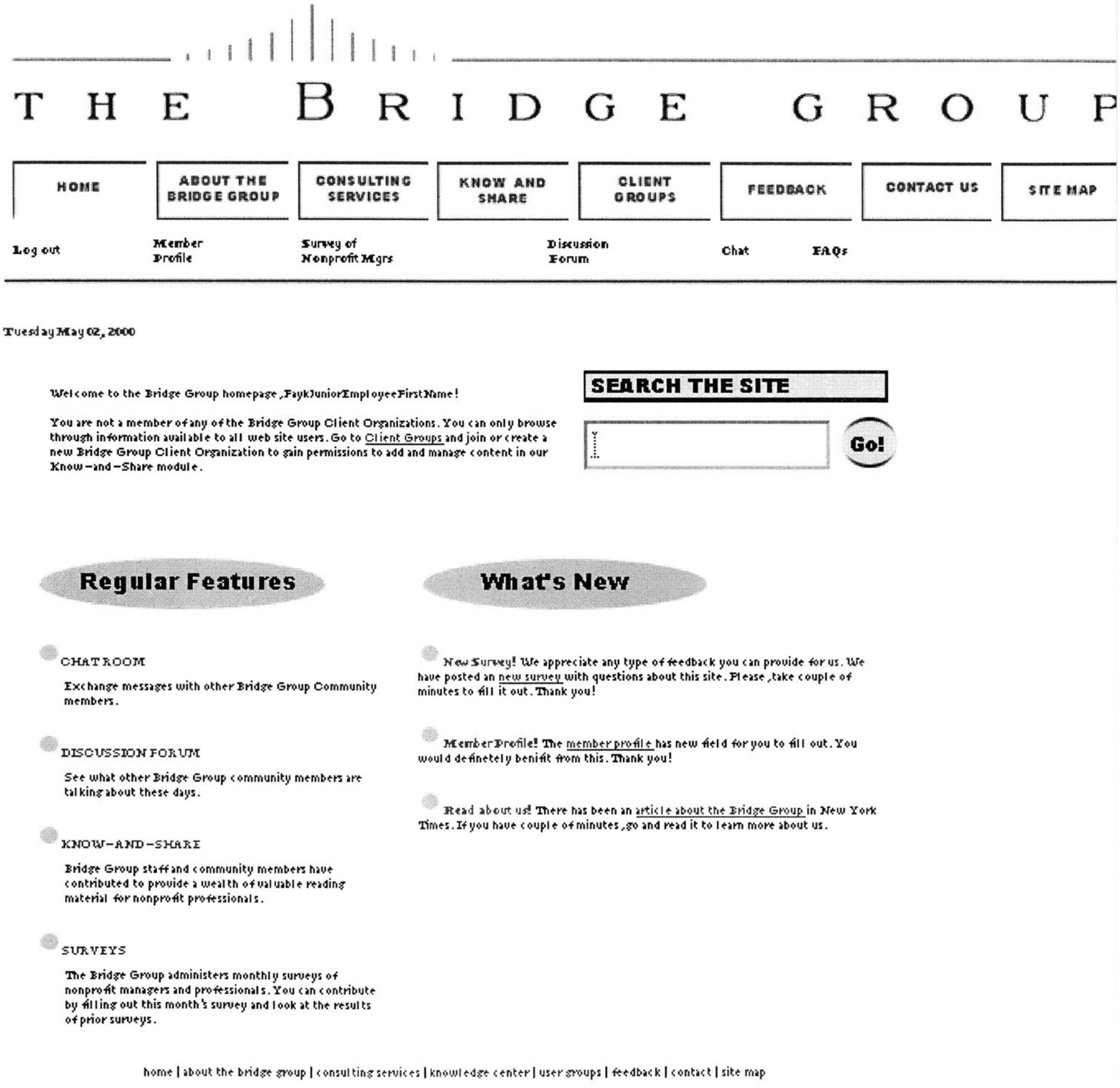
To complete his task, FaykFrequentUser follows the "Find Similar Objects" link and sees that one of the cases that has been posted by the BridgeGroupEmployee mentions the government agency FaykFrequentUser spoke about in his proposal. FaykFrequentUser provides a public link from his proposal to the case.

2.5 KGBFrequentUser

KGBFrequentUser is an employee of KGB Nonprofit Organization. He frequently visits the site and contributes new content. KGB Nonprofit Organization also works very closely with FaykFirm on a few projects. Thus, KGBFrequentUser decides that this site could be useful for these two organizations to share knowledge. As an example, he decides to create KGB-FaykFirm Conference Notes. He goes through the process of creating a new knowledge type and new objects (as describe above) and sets appropriate permissions for employees of the FaykFirm and KGB Nonprofit Organization. Thus, the employees of both organizations are ready to use the KGB-FaykFirm Conference Notes: they can create new Conference Notes objects; they can edit new Conference Notes; they can comment, rank, bookmark, and link them to other knowledge objects.

Note that this scenario demonstrates the capability of Know-and-Share to handle inter-organizational knowledge sharing. If a user is not a member of either organization, he/she cannot use this object type because he/she does not have appropriate permissions. Thus, the security is not compromised, and employees of both organizations can freely share proprietary information with each other. At the same time, some proprietary information (e.g., Meeting Notes within FaykFirm), is visible only to one organization, the FaykFirm.

The scenarios above briefly describe some of the features that are available to the Bridge Group clients through the Know-and-Share Center. The Design and Implementation sections of this document will go into each of the features in more detail.



Tuesday May 02, 2000

Welcome to the Bridge Group homepage ,FaykJuniorEmployeeFirstName!

You are not a member of any of the Bridge Group Client Organizations. You can only browse through information available to all web site users. Go to Client Groups and join or create a new Bridge Group Client Organization to gain permissions to add and manage content in our Know-and-Share module.

CHAT ROOM

Exchange messages with other Bridge Group Community members.

DISCUSSION FORUM

See what other Bridge Group community members are talking about these days.

KNOW-AND-SHARE

Bridge Group staff and community members have contributed to provide a wealth of valuable reading material for nonprofit professionals.

SURVEYS

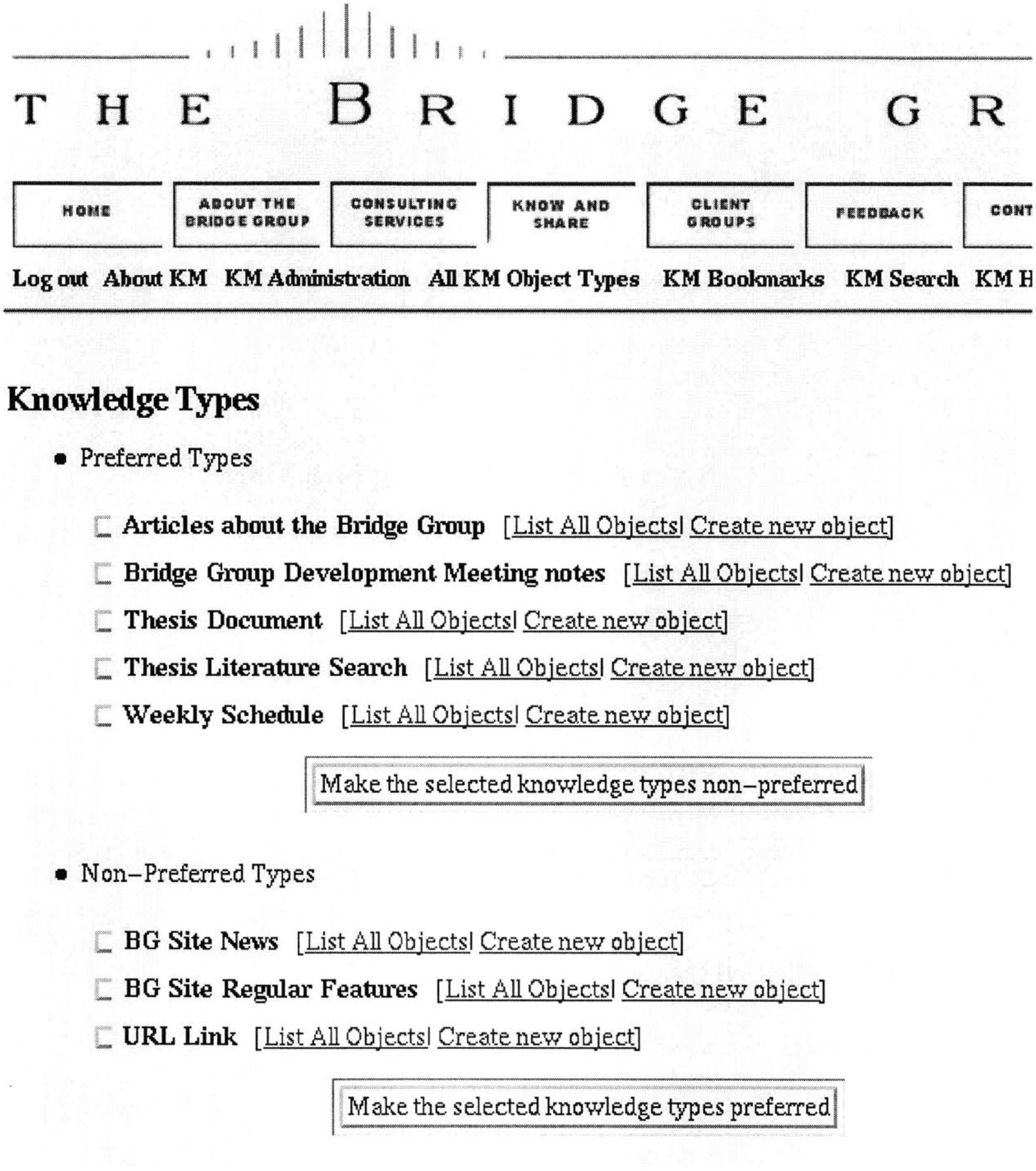
The Bridge Group administers monthly surveys of nonprofit managers and professionals. You can contribute by filling out this month's survey and look at the results of prior surveys.

New Survey! We appreciate any type of feedback you can provide for us. We have posted an new survey with questions about this site. Please ,take couple of minutes to fill it out. Thank you!

Member Profile! The member profile has new field for you to fill out. You would definetely benifit from this. Thank you!

Read about us! There has been an article about the Bridge Group in New York Times. If you have couple of minutes ,go and read it to learn more about us.

Figure 2-13: Main Page of the Bridge Group Site

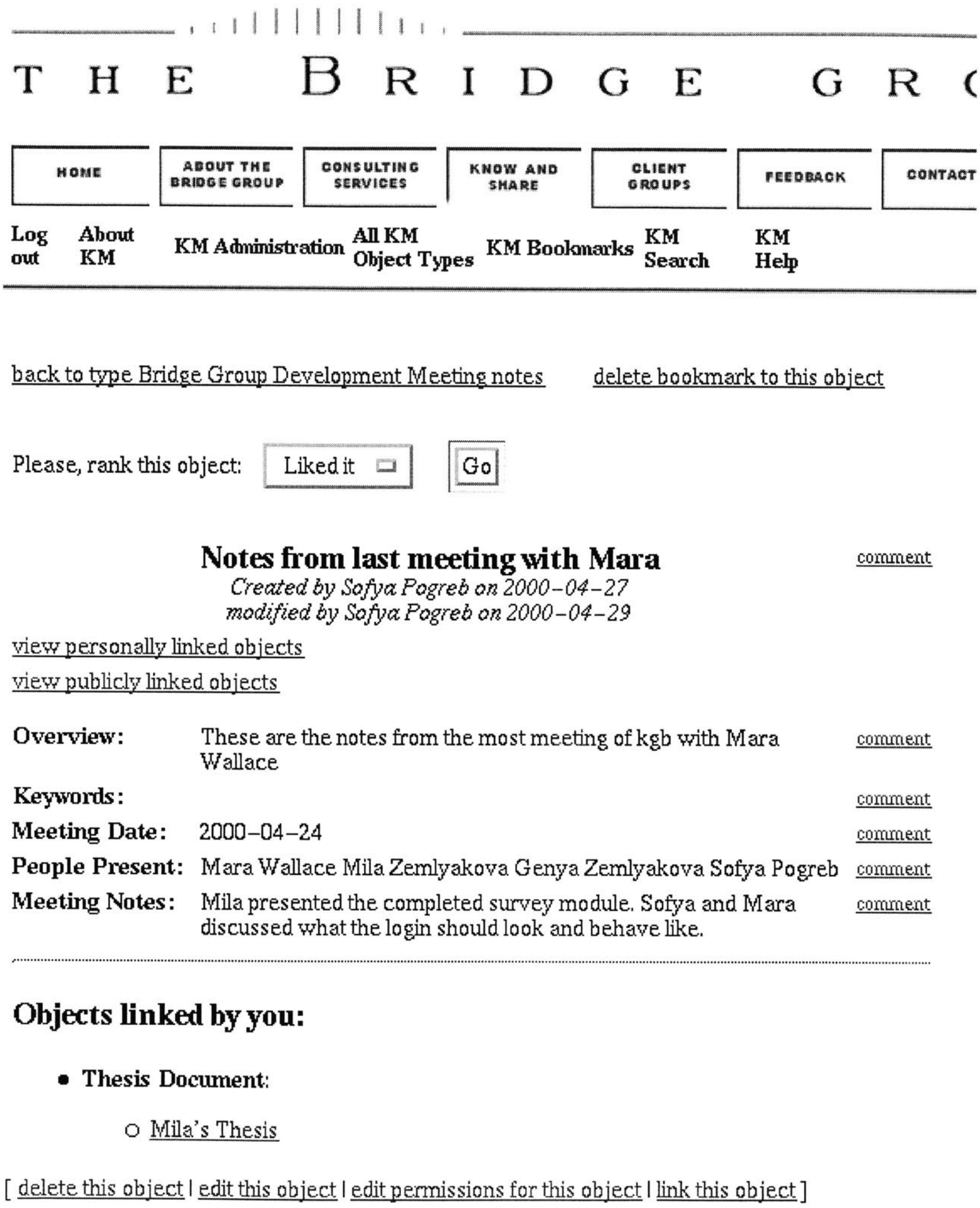


Knowledge Types

- Preferred Types
  - □ **Articles about the Bridge Group** [List All Objects| Create new object]
  - □ **Bridge Group Development Meeting notes** [List All Objects| Create new object]
  - □ **Thesis Document** [List All Objects| Create new object]
  - □ **Thesis Literature Search** [List All Objects| Create new object]
  - □ **Weekly Schedule** [List All Objects| Create new object]

- Non–Preferred Types
  - □ **BG Site News** [List All Objects| Create new object]
  - □ **BG Site Regular Features** [List All Objects| Create new object]
  - □ **URL Link** [List All Objects| Create new object]

Figure 2-14: Main Page of Know-and-Share Module



Please, rank this object: Liked it

Notes from last meeting with Mara

Created by Sofya Pogreb on 2000-04-27
modified by Sofya Pogreb on 2000-04-29

view personally linked objects

view publicly linked objects

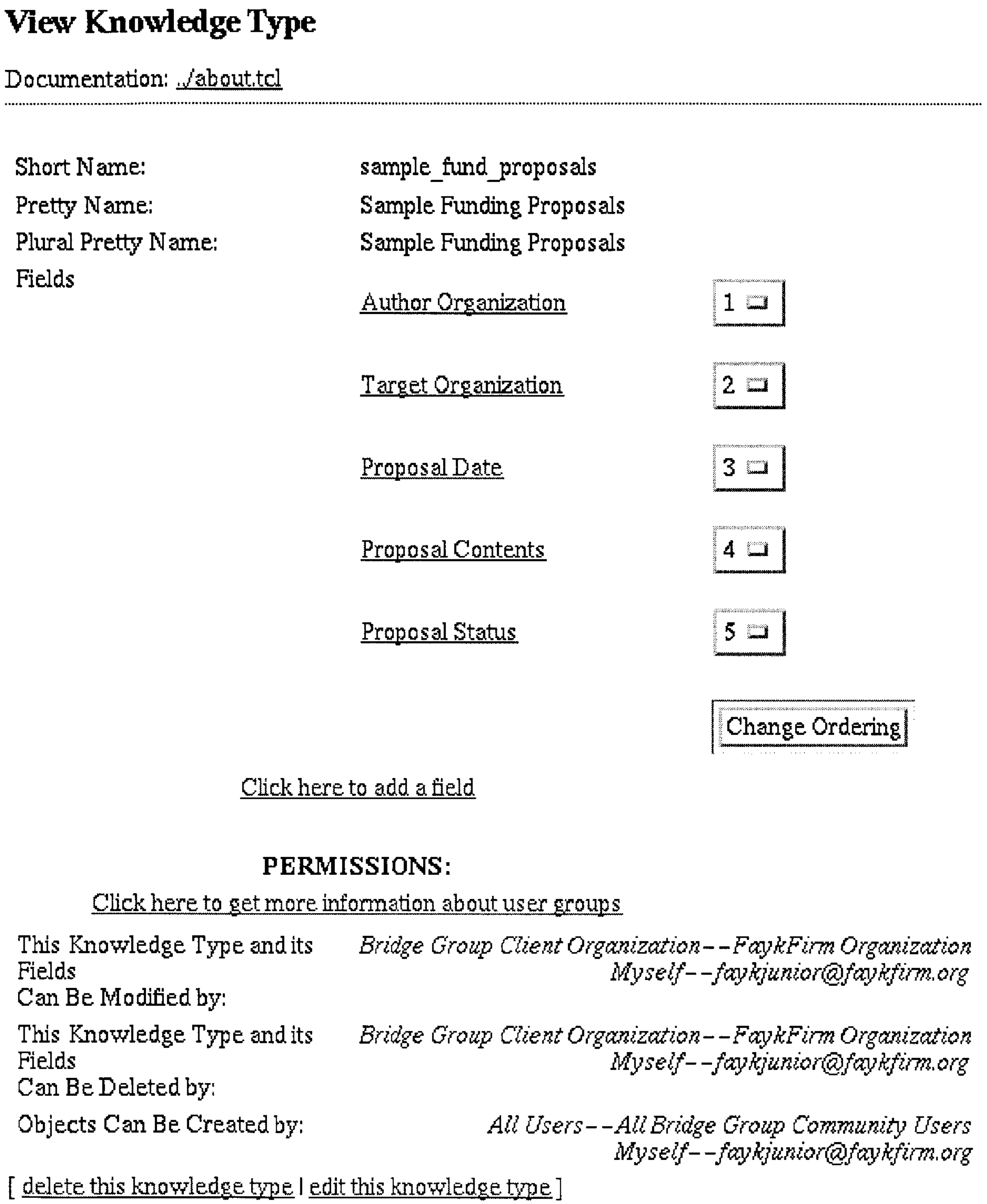
Overview:	These are the notes from the most meeting of kgb with Mara Wallace	comment
Keywords:		comment
Meeting Date:	2000-04-24	comment
People Present:	Mara Wallace Mila Zemlyakova Genya Zemlyakova Sofya Pogreb	comment
Meeting Notes:	Mila presented the completed survey module. Sofya and Mara discussed what the login should look and behave like.	comment

Objects linked by you:

- **Thesis Document**:
  - Mila's Thesis

[delete this object | edit this object | edit permissions for this object | link this object]

Figure 2-15: Main Page of Know-and-Share Module



View Knowledge Type

Documentation: ../about.tcl

Short Name:	sample_fund_proposals	
Pretty Name:	Sample Funding Proposals	
Plural Pretty Name:	Sample Funding Proposals	
Fields		
	Author Organization	1
	Target Organization	2
	Proposal Date	3
	Proposal Contents	4
	Proposal Status	5

Click here to add a field

PERMISSIONS:

Click here to get more information about user groups

This Knowledge Type and its Fields Can Be Modified by:	*Bridge Group Client Organization--FaykFirm Organization* *Myself--faykjunior@faykfirm.org*
This Knowledge Type and its Fields Can Be Deleted by:	*Bridge Group Client Organization--FaykFirm Organization* *Myself--faykjunior@faykfirm.org*
Objects Can Be Created by:	*All Users--All Bridge Group Community Users* *Myself--faykjunior@faykfirm.org*

[delete this knowledge type | edit this knowledge type]

Figure 2-16: Page that Displays the Object Type's Information

Chapter 3

Related Work

Work that has been done in the field of knowledge management includes both published research articles that explore the abstract and concrete ideas for knowledge creation and sharing and commercial systems that try to exploit some of these ideas and provide software tools for managing information.

This chapter first describes the scientific and commercial publications produced in the last few years in the field of knowledge management and summarizes concepts and features that are essential for a successful system. The second part of this chapter presents a few commercial systems that are available on the market, describes their features and main ideas behind creating them, as well as the reasons why they are not adequate for the goals faced by organizations such as The Bridge Group.

3.1 Literature Overview

3.1.1 Fundamental Ideas

Many articles discussing ideas behind knowledge creation, sharing and management compare and contrast the evolution of knowledge to basic biological systems and organizations. For example, while the evolution of knowledge is similar to biological evolution and has the same requirements (variation, reproduction and spread, and selection), at the top level it is different from the biological systems in the following ways [1]:

- "Objectification". People's ideas and experiences need to be converted into discussable objects. When some flexible structure is imposed on these objects, they become easy to find, evaluate and modify by several people. Some of the ways to achieve objectification include "putting these ideas and experiences in writing" and make it available to a large community of users (which can be achieved through Internet technologies).

- "Behind-the-scenes" views. Usually, the knowledge that goes into a final product is not available for reuse, criticism and modification. A successful knowledge management system should find ways to display the process of creating knowledge or creating flexible document structures that can facilitate individuals and teams of people in developing methodologies, project evaluations, etc.

- "Outbreeding". Using the biological evolution analogy, knowledge evolution should be aided by mixing knowledge objects from different user groups and organizations. The World Wide Web already provides some ways to accomplish the latter goal, for example, discussion forums, chat rooms, etc. However, the system that deals with a problem of effectively managing information has to provide ways for groups of people to feel that they are connected with other groups and can leverage knowledge and information created and maintained by other groups. These groups should have options to share certain knowledge with other groups, and at the same time, protect certain information that is proprietary.

The systems that will be discussed later in this chapter address some of these top-level issues. Most of them do an adequate job of structuring different knowledge types, so that they can be easily displayed and stored. However, the structures imposed on these knowledge types are very rigid and modifying them requires a great deal of effort. Thus, these systems achieve the goal of "objectification", but fail in solving the problem of "behind-the-scenes" views, which, in general, require a more flexible knowledge organization. When addressing the third goal, "outbreeding", most designers of knowledge management systems provide standard virtual community features, such as discussions and chat, but don't encourage, or even provide means for inter-group or inter-organizational collaboration. Thus, most KM systems exist for a closed group of users who are mostly sharing knowledge among themselves.

3.1.2 Concrete ideas

The design and implementation of the knowledge management system will depend heavily on the shape and strategic drivers of an organization. Therefore, the creation of such system within a specific organization should be aimed at adding value to the organization, its employees and clients. In the literature, a few common objectives, which have to be addressed by designers of these systems, have been identified:

- Promoting creation of intellectual property. This goal concentrates on creation, maintenance and expansion of the main knowledge base of an organization. For example, for a management consulting firm, it might consist of client presentations, case studies, articles, and methodology documents.

- Customer privacy. An effective system provides an elaborate authorization scheme that allows users (or clients) to create knowledge for personal or group use, as well as sharing it with the rest of the world. It shouldn't be possible to track pseudonymous or anonymous authors without administrative privileges or publish something under another's unique name or pseudonym [3].

- Promoting innovation and process/product quality. This objective concentrates on obtaining information about experiments and experiences of different individuals and groups within or outside of the organization. A knowledge management system, the goal of which is to improve the process and product quality within an organization will place heavy emphasis on collaboration, sharing of ideas and best practices.

- Promoting rapid response and evolution of knowledge. Any knowledge management system has to support collaboration between users and groups of users. The ways to encourage collaboration include supporting effective criticism, providing ways to share information, and supporting flexible filtering [3]. In particular, for Web-based systems, meeting these objectives will result in creating links between different documents, as well as a way to comment on specific documents.

- Promoting personalization. The state-of-the-art systems running on the Internet that gather and filter information provide many ways for users to personalize their experiences. Designers of these systems have developed different approaches to learning user behavior through positive and negative user's relevance feedback. Since selecting relevant information for a particular user or user group might be a time-consuming process, "recent efforts have been devoted to overcome this problem by personalizing an information filtering system" [4]. Such systems take into account user's feedback and present relevant information to users more effectively.

It is apparent that some of these goals overlap or even conflict with each other. The biggest problem that most knowledge management system designers face is how to leverage one or more of these goals that are crucial to a specific organization or a community without being too limited on one hand, and too overwhelming and inefficient on the other. This problem becomes even more challenging when the system is designed to be used by multiple organizations and has to satisfy requirements of different groups of users, which in some cases might be completely opposite, and at the same time bringing multiple groups together and promote inter-group collaboration and communication. As the next section of this chapter shows, most existing systems target a specific group of users and do not allow extensive collaboration and knowledge sharing among different groups.

3.2 Systems

This section presents a few systems which have similar capabilities as Know-and-Share, describes their features and concepts behind them and compares them with Know-and-Share. Unfortunately, most of these systems, with the exception of Intranets.com, are commercial and allow only limited access to its features. Thus, our evaluations are based on their descriptions and publicly available demos and evaluations. Some of these systems provide a wide variety of features, which can be very useful to the users of the system, but we will concentrate only on the features and ideas that are most relevant to this thesis.

3.2.1 Intranets.com

An Intranets.com site gives users a free package of popular intranet applications accompanied by an intuitive interface. Any group of people can set up their own Intranets.com intranet [10]. Each member of this group will need a computer with internet access, and the intranet site which is accessible from any computer. The site is password protected, and the information created and managed by this user group is available only to members of this group. The following communication capabilities and business services are provided for free to all intranet users: document management, group calendar, group discussions, member directory, contacts directory, instant messaging, announcements, customized news and financial information.

In order to register a group and create an intranet site, the administrator of the newly created site has to go through four different forms, which ask for a lot of mandatory information, such as the industry information, personal addresses, phone numbers, etc. Once the site is created, people can join the group and start using the features offered by Intranet.com. The features that are most relevant to this thesis are the document management module, group management and communication modules such as instant messaging and group discussions.

The document management module has very limited capabilities. It only allows users to upload files and create folders for better management of uploaded files. All the files uploaded by users of a particular intranet site are visible to the entire group and can be modified and deleted by any member of the group. In order to modify a file, the user has to download the file, modify it locally and then upload it again into the system overriding the old version. Thus, besides the fact that no personal documents can be stored in the system, a file that is meant to be viewed by all members of the group can be inadvertently modified or deleted by a member of the group not familiar with its content. This will prevent users from sharing important documents with other members of the group if they want to prevent modification of these documents, thus severely limiting the users' contribution to the information present in the system. All of these capabilities are currently present in the Know-and-Share system with an addition of a authorization scheme that allows users to select which community members are able to modify and view the files. An even bigger disadvantage of the Intranet system is that it doesn't allow creation and modification of knowledge online, which is one of the core functionalities of the Know-and-Share.

In general, the Intranet sites do not provide any kind of authorization for information stored or displayed on the site. No private or semi-private information (which can be shared with a subset of members of the group) can be posted on the site.

Communication among the Intranet users is encouraged through the use of the Discussion Forums and Instant Messaging (which are equivalent to the Know-and-Share discussion forum and chat modules). This is definitely an important feature in a knowledge management medium.

Overall, the Intranet system is designed to be used by a relatively small group of users who need some way of storing shared files and communicating with each other online. An Intranet site is a closed environment, in which different groups of users are not integrated with each other and cannot share or collaborate on any types of information. In contrast, the Know-and-Share system encourages the reuse of knowledge and collaboration among different organizations, while at the same time providing the level of privacy desired by every organization.

3.2.2 SiteStation

SiteStation is a Web document and knowledge management system that provides "a unique Web content viewing experience based upon user profiles" [12]. It allows an organization to create a searchable dynamic Web site that can be maintained by non-technical staff. SiteStation lets the originators of content post articles with start and expiration dates and associate images with the content. Since all content is managed, it lets the authors control who can edit and who can read it. If a user doesn't have rights to the content, he/she will not see links to that information. SiteStation, as most existing knowledge management systems, does not allow online editing of documents. In order for an authorized user to modify a document, he/she has to:

1. download it to his/her local computer for editing

2. modify the document in its native application

3. save the file in its native application to the local disk or network drive
4. upload the file into the SiteStation system

This is a long and cumbersome process, the main disadvantage of which, besides taking long time for the editing process, is that both the author and the readers have to have "the native application" in which the article has been written. Also, it is virtually impossible for the administrator of the site to impose and reinforce any sort of structure on a specific type of knowledge. For example, there is no mechanism to enforce a content originator to include a specific fields in his/her article, except for a written instruction asking the user to do so. Moreover, people browsing the documents don't have a uniform way of viewing similar information pieces.

All content in the system is stored in the SiteStation database, and links are controlled by the system, so that when the content is deleted or expired, users don't see "dead links" to the old information. Content is organized into folders, and a single article can appear in more than one folder. Content security is managed at the folder level. All documents that appear in a folder inherit the security restrictions of the folder.

In the system, the user can be a member of more than one group, and his/her security restrictions are inferred from the permissions given to one or more user groups he/she belongs to. Those articles that he/she is not authorized to edit or view are not displayed during both browsing and searching.

In general, this system has a carefully designed authorization architecture, but fails to take advantage of online knowledge development, such as modifying content online. Also, one cannot personalize the way a user is presented with information. For example, a user cannot choose which kind of knowledge he/she is interested in. He/she is forced to see all the folders and all the documents he/she has access to. Additionally, there is no mechanism for the user to jump to a particular type of information. Both of these mechanisms exist in the Know-and-Share in the forms of links and bookmarks. SiteStation also doesn't provide any way for the users to comment on specific objects (for personal user or provide criticism and suggestions to other users) or even collaborate online.

3.2.3 Novation Products

This section describes a typical content management application that is designed to be used by a large organization, needs a great deal of customization and tuning up to suit the needs of a particular organization, and requires a lot of effort to maintain [14]. Novation is one of the most complex and elaborate desktop applications that exist on the market today. It is a data management application developed by Cimage NovaSoft. It provides a rich feature set which includes:

- Content and Document Management. The content management part of the application has been very carefully designed and oriented toward heavy collaboration among originators of the content. Novation supports many types of documents, including customized and elaborate data entry forms, which have to be created using the Novation's Forms application. All information that is stored in the system is version controlled, tracked, secured and available as the Web-readable content. Users can view the same content in different formats on the Web or in their native applications. All documents are stored in multi-level folders, which provide an easy way to manage multiple content types.

  The ability to impose a structure on a document can be controlled by the administrator. For example, required attributes can be defined for each document type, or checks can be performed on the entered data.

  The document creation follows a workflow approach. A project leader or a manager is responsible for starting a workflow, which includes the creation, editing, approval, reviews of documents. The following is an example of a typical workflow: James creates document A, Ann edits field1 in document A, John edits field2 in document A, James approves and publishes document A.

- Access Control. The Novation's security model covers the following areas: document, folder, user level and group level security. The application is also password protected and uses electronic signatures to protect sensitive data. These levels of security can be applied together or independently.

- Distributed Document Management. To promote and encourage collaboration, the Novation's architecture provides the users with a secure way of reaching authorized users almost immediately. For example, if some content needs to be developed by more than one user, the software provides an easy way to transfer control among users.

- Novation Search. The software offers two types of search facilities: the content-based retrieval searching and table searches of document attributes. Users can save the results of the search or access searches created by others.

- Integrated Graphical Workflow and Electronic Forms. Although these features can be very useful within an organization, they are not essential to creating a successful knowledge management system and not relevant for the purpose of this thesis.

The list of features provided by the Novation is very impressive. However, its biggest disadvantage is that it requires a lot of hardware and software installation and configuration by the Novation customer support. Novation uses a component "assembly" methodology, which means that before deploying the system within a specific organization, a team of developers has to customize and change the product using Visual C++, Java, Visual Basic or other Java or COM component-based tools. In general, this system is resource intensive and can probably be maintained only by a large organization. Since most Novation products are desktop applications, every content originator in the system needs to install Novation software on his local or network drive. Thus, the system fails to exploit the power of the Web as the knowledge development media. Moreover, the software itself is very complicated (including its user interface), thus content administrators require extensive training.

Although systems like Novation try to promote collaboration, the workflow model is very rigid. Only project managers can create work flow for creating and updating documents and define the order in which users modify the document. The manager fully controls the document development, and any sporadic changes to the knowledge objects require him/her to update the work flow manually. For example, if the original work flow is "Start → James creates document A → Ann edits field1 in document A → John edits field2 in document A → James approves and publishes document A → Finish", Ann cannot edit the document after James published the document and the work flow reached "Finish". She has to contact the project manager who has to update the work flow for this particular document to allow her to modify the content. Thus, Novation document management is targeted towards creating high quality finished pieces of content that are published after they have been carefully developed. It is not very efficient when the evolution of knowledge requires user feedback (i.e., modification of content after it has been published). This kind of control over the development of knowledge is mostly suitable for a structured business environment, and would be very restrictive for a community of Web users especially if these users belong to multiple organizations who want to create and manage pieces of knowledge in a weakly structured fashion.

Overall, the system lacks the look and feel of a community. It doesn't provide any utilities for users to freely communicate ideas to each other (e.g., comment on different pieces of information or provide feedback), or edit content in an unstructured manner (e.g., after the workflow process is done, in order to modify the content again, the administrator has to create another flow).

3.2.4 Xpedio Content Publisher and Content Server

Another set of knowledge management applications in the market focuses on leveraging the power of the Web and tries to provide the Website administrators with a set of tools to define the look and feel of the Web content developed by the users of the system. An example of such system is developed by IntraNet Solutions, which consists of Xpedio Content Server and Xpedio Content Publisher. Overall, this system lets organizations "rapidly deploy content-centric enterprise Web applications and supply Web sites and other applications with consistent, managed content" [15]. It also provides a rich set of content publishing tools that help Website administrators create and manage enterprise Web sites. The system allows users to contribute and access information using a Web browser.

One of the key features of the system is the ability to publish and manage content directly from the source with which it was created. Content originators can develop their documents in their native applications, such as word processing and spreadsheet software. Then Xpedio Content Publisher converts those documents into linked Web sites using template-based technology without modifying the source documents. In order to modify the document, the author modifies it in its source document and Xpedio Content Publisher automatically updates all reference documents and navigation links. Also, the author can choose the dates when his content appears on the site and when it expires.

To facilitate navigation, a number of useful additional features is provided, such as creating tables of contents, lists of figures and tables, reports and keyword indexes. Additionally, the content can be personalized for the individual user. He/she is provided with navigation capabilities customized for him based on his permissions and interests. Thus, users do not see links to the documents that they are not authorized to view or modify or to the irrelevant information not related to their job.

Although the system provides an easy way to create and modify knowledge on the web, it places a great deal of responsibility on the webmaster, who manages the site. The webmaster uses software, called Site Builder to create project files, which describe the layout for each type of content. Moreover, the initial development of the site can be challenging as the Site Builder is not very intuitive. The content is delivered to the webmaster through email-based workflow. Also, the website administrator sets the rules for the templates and structure of different knowledge types.

One of the problems not addressed by the system is the collaboration among different users. A person who wants to modify a piece of content needs physical access to the native application in which the content was developed. Again, like in other knowledge management systems, the content marked for editing has to be downloaded to the local desktop and then uploaded back to the server when it is checked in.

Even though the Xpedio products strive to provide users with personalized content, there are no mechanisms for efficiently bookmarking, rating, or commenting on the content produced by the users.

In conclusion, most systems available in the market today try to solve the problem of knowledge management by concentrating either on providing an extensive set of content management and presentation tools (such as Novations) or creating a community of users who can easily communicate different ideas to each other and have easy access to the documents published by other members of the group, but don't give tools for efficient editing, posting and viewing content (such as Intranets.com). In both cases, these systems don't support inter-organizational communication and collaboration. In order to share the content among different organizations and encourage them to leverage each other's experiences and knowledge and collaborate on the creation of new content, either additional software has to be installed for every user group, and administrator of those groups have to be extensively trained, or the process of inter-group collaboration is so cumbersome and inefficient, it is not worthwhile for multiple organizations to share the same content management system. As the rest of this thesis shows, the Know-and-Share system supports both an easy process of creating, modifying and sharing content and collaboration among various organizations, pursuing different goals and priorities.

Chapter 4

Goals

Knowledge Management systems are not a novel idea - in fact, most large organizations have proprietary Knowledge Management systems. A typical system requires a team of several technically savvy full-time employees dedicated solely to running the system. Such a system may offer users ability to search and browse the collection of documents assembled by the administrative team. Adding to the collection usually involves submitting the new document to the system development team which then reviews it and, in case of approval, adds the document to the system and thus makes it available to the users.

Those rare Knowledge Management systems that allow users to directly enter content usually only accept several mainstream document types – for example, they may accept articles and cases. However, if a user feels that a Research Paper document type would be a useful addition, he/she has to contact the administrative team, convince them a new type is indeed necessary, and then wait till the programming required to add a new document type is completed.

The limitations described above discourage users from contributing to the knowledge base, confining them to the role of passive observers. Thus, Knowledge Display seems to be a more accurate definition of what usually claims to be a Knowledge Management system.

The goal of the Bridge Group Community is to facilitate collaborative creation, navigation and collaborative development of knowledge by two or more user entities, where an entity may represent an entire Client Organization, a project team within a Client Organization, or a single employee of a Client Organization. The Know-and-Share Center, which is the core module of the Bridge Group Community, will aim to enable sharing of information and allow users to contribute to the material created by others. The Bridge Group is a nonprofit organization and is not thrilled about the idea of dedicating several full-time employees to the sole task of managing the

Community. Thus, the Know-and-Share Center has to require a minimal amount of administrative resources. Also, most Bridge Group employees have limited programming skills. Therefore, adequate utilities should be provided by the system to make it feasible for a non-technical person to perform the small amount of administrative work that is required. A person with the knowledge of technology will obviously be needed to attend to events such as bug reports, but one of the primary goals of the Bridge Group Community is to make the day-to-day task as non-technical as possible.

While ease of administration is required for the Bridge Group to use the system, it will not do anything to persuade the Bridge Group Clients to use the site. Thus, it is crucial that minimizing the amount of administration required should not come at the expense of capabilities the system offers to the users. Users should be able to directly enter content, modify content contributed by others, browse and search the knowledge. They should be able to leverage the existing structure of the Know-and-Share Center by reusing the Knowledge Types already in place, as well as to extend the structure of the Know-and-Share Center by introducing new Knowledge Types. Making the Know-and-Share Center extensible is very important: the content has to be created somehow, and giving the users the power to develop it accomplishes two fundamental goals – it provides a faster, more direct way for the user to contribute to the content and, at the same time, leaves less work for the administrator.

A versatile authorization model is one of the most important goals of the system – the community must be user-group-oriented: it is intended for collaboration among organizations and should thus provide an extensive framework for assigning and managing permissions. Authors should have the options of making knowledge available to particular users as well as groups of users. Thus, the system should allow assigning permissions to an arbitrary subset of users. Client Organizations must feel that they can make their content as public or as private as they wish – otherwise, they will not use the system.

In addition to tools for creating content, utilities for personalizing the content navigation experience according to the user's interests should be available as well. Finally, the Bridge Group Community should include modules to help enhance the usability of the Know-and-Share Center. A discussion forum and online chat features should be provided. The site should look and feel like a true community, offering interesting content. It should also offer ways to manage, create, and share that content, ways to discuss, evaluate, and rate it – it should provide a complete integrated experience.

Chapter 5

System Design

A successful knowledge management system that meets the goals described in the previous section has to efficiently manage knowledge with the required level of security, facilitate the evolution of knowledge in the system, and at the same time provide enough personalization to allow every user of the system to find most relevant information. This chapter focuses on the design of the system and gives an overview of different features and functionalities provided by the Bridge Group Community system and provides rationale behind different design decisions.

5.1 Representation of Knowledge

Unlike other knowledge management systems that require new knowledge to be submitted by uploading a file, Know-and-Share supports form-based creation and editing of knowledge. While calling for a more extensive user interface than would be required otherwise, supporting form-based content management has several important consequences:

- The only application users need to work with a form-based document is a web browser. This is especially significant given that the system is targeting non-profit organizations whose employees often use equipment that leaves a lot to be desired. Another implication is the fact that the operating system the users are on will make no difference -- for example, if uploading was the only way of submitting information, many documents would probably be in Microsoft Word format. Unix users would then have problems using and editing the content.
- Content can be viewed and modified online. The user does not have to download the document in order to view or modify it. This, once again, is important

because the system will have to accommodate users with limited resources. A user who has to wait for 10 minutes to download an article he wants to skim is not a happy user.

- The system can impose some structure onto the content. In a file-upload-only knowledge management system, the only thing that differentiates an article from a funding proposal is the fact that the respective authors designated the first to be an article, and the second a proposal. Know-and-Share lets users create Knowledge Types, defining each type of knowledge by its structure. Thus, an article maybe a document that consists of a title, author, date of publication, abstract, body, and references. Structuring content makes knowledge more uniform, systematically organized, and easily searchable making the user's experience less sporadic.

The system will thus represent content as different types of knowledge. This design choice calls for a mechanism to support classification of documents according to their Knowledge Types, as well as utilities to create and manage Knowledge Types. Each newly created Knowledge Type is translated into a dynamically created Knowledge Object table which will eventually contain objects of that Knowledge Type.

Content of each type consists of several parts (for an article, the parts may include title, author, abstract, etc.). These parts are modeled as Knowledge Fields and correspond to columns in the dynamically created Knowledge Object table which is created when a new Knowledge Type is added. Each Knowledge Field has a number of attributes, including:

- the kind of data the field is expected to contain (i.e. title should be a string)
- how the data should be presented for editing (i.e. days of the week should be presented as a dropdown list)
- whether the field is mandatory (i.e. should the author field necessarily be filled out for each article)
- where the field should be displayed relative to other fields of the object (i.e. the article title should probably be above its abstract)
- default value for the field that is assigned if the user leaves the field blank
- entry explanation that is used to prompt the user to enter information

All Knowledge Types have several common system-defined Knowledge Fields such as Name, Overview, Keywords and Created By. New user-defined Fields can be added at any time –– objects of the Knowledge Type that have already been created will have the corresponding field set to the default value if the new Knowledge Field has one, or will be empty otherwise. Knowledge Fields can also be deleted –– in that case, the field data from the already existing objects is lost.

Finally, a Knowledge Object is modeled as an entry in a dynamically created table corresponding to the Knowledge Type the Object belongs to. An Object will have a value corresponding to each Knowledge Field of its Knowledge Type (some of the values may be null). Thus, each Object will have all of the system-defined fields mentioned above, enabling the system to ensure all Objects have data on who created them, and each Object has a name and an overview (necessary for listing and displaying objects in a coherent fashion). Each object also has a Keywords field the creater of the object can fill out. The Keywords field is used to search for similar Objects (the search feature is described in more detail in a further section). Once a new Object is created, it can be modified and deleted. Therefore, the content of the Know-and-Share Center is not static –– the documents can be edited and no-longer-needed ones can be removed from the system. In general, all objects are physically removed from the system to save space as opposed to marking them as being deleted and storing them in the database indefinitely (although this decision might be reconsidered in the future if the need arises).

Representing knowledge in a way described offers the users maximum extensibility and encourages continuous evolvement of the content. The design choices described above provide the necessary dynamic environment for fast creation and management of different knowledge types and objects. The presented model of creating and managing knowledge types is radically different from many systems that exist on the market because it allows dynamic changes to the knowledge types through the Web. In other systems such changes have to be made either by a technical person who is familiar with the programming language and database management or by an administrator of the site through the use of special software.

It also makes it crucial for the system to provide a fine-grained authorization architecture so that the user who creates a knowledge object can specifically define the subset of users who should be able to perform various tasks (such as viewing, modifying or deleting) on the document. The design of the Know-and-Share permissions is described in a later section of this chapter.

5.2 User and User Group Management

The Bridge Group Community is a user-group-centered system. The Know-and-Share Center is intended to facilitate sharing of content between groups of users. This makes it unlike the typical Knowledge Management systems which are intended for sharing content within a particular group, thus effectively being Intranets. Know-and-Share has to be both an Intranet and an Internet since there will be content intended for sharing with other groups and content that is perceived as proprietary and intended for internal sharing.

Although Know-and-Share revolves around user groups, some of the features that help the user manage the content more efficiently are user-oriented rather than user-group-oriented. These features will be discussed in more detail later, but an example of such a feature is bookmarking an Object. A bookmark is personal and is only visible to the user who created it. Features like bookmarking require each user to have a distinct identity. Another reason for user identities is the need to be able to identify the user that created/modified a particular object in case questionable content is posted. Finally, without distinct user identities, the system would not be able to assign different roles to members of user groups.

To become a full-featured user of the Know-and-Share center, an individual has to join one of the Bridge Group Client Organization groups. Joining a Client Organization group requires Group Administrator approval, thus ensuring that only members of Client Teams gain rights to create and modify content. The first member of a particular Client Organization to apply for Know-and-Share membership has to fill out a New Client Organization Group form. The creation of the new Client Organization Group requires approval by a member of the Bridge Group Employees Group who can verify that the user does, indeed, represent a Bridge Group client. After obtaining approval, the first member of the Client Organization automatically becomes the Client Organization Group administrator and becomes responsible for approving or denying membership to users who attempt to join the group from then on. If the Client Organization feels the member who happened to apply first is not appropriate for the administrator role, a different administrator can be subsequently assigned.

Passing the responsibility to approve new members on to the Client Organization Administrator serves a dual goal. First, the Client Organization Administrator is likely to be more aware of whether a particular user is employed by the Organization (the Bridge Group employees may not have an extensive list of all people working on a project). Second, putting each Client Organization Administrator in charge of his/her organization reduces the work load of the Bridge Group Site administrators who would otherwise be responsible for all Client Organization Groups –– a burden that is a violation of one of the system's primary goals –– minimizing administrative requirements.

A user may choose to be a member of more than one Client Organization Group given that he is approved by the administrator of each group. This capability is provided for the purpose of accommodating an unlikely but possible situation where a user is employed by more than one Bridge Group Client Organization.

In addition to these top-level Client Organization groups, the system offers its users an opportunity to create other user groups (membership in which may or may not require the administrator's approval) and share knowledge with other members of those groups. The decision to differentiate between client organizations and other groups was made in order to avoid overwhelming users with a large number of different groups (e.g., when setting permissions) and at the same time, to provide enough security and control over the client groups' membership.

Individuals who are registered users of the Bridge Group community but do not belong to a Client Organization group can browse content which is designated as viewable by All Website Users. The latter design decision was to accommodate potential Bridge Group clients who may want to browse the Know-and-Share Center to see what the site offers.

5.3 Security and Authorization Model

An authorization architecture that is both flexible and clearly defined is essential in a system that aims to provide a variety of sharing options. It is important to convey to the users that the content they create will be under their complete control -- they can do as much or as little sharing as they deem appropriate. If the user is not comfortable with the security architecture, he/she won't be comfortable with the system.

There are two levels of authorization in the system -- at the Knowledge Types and Knowledge Objects level. This design choice provides enough security to protect proprietary information, on one hand, and avoids an unnecessary burden on the authors to deal with too fine-grained permissioning options.

5.3.1 Knowledge Type Authorization

When a new Knowledge Type is created, it becomes available to authorized groups of Know-and-Share users. This design choice gives generators of Knowledge Types enough control over the structures they create, so that if they decide to include any proprietary information into the structure of Knowledge Types (e.g., provide some default information for different fields within the types or include confidential information into the select boxes), they can prevent unwanted access to these knowledge types. The downside of this choice can be illustrated by a situation when some common types are not available to all system users, and this will force different user groups to create similar Knowledge Types, thus not reusing the ones that already exist in the system. For example, the system may contain dozens of different Article Knowledge Types. The choice between the two alternatives (restricted and unrestricted access to knowledge types) was made based on the desire of the Bridge Group to have a greater level of security with regard to knowledge types.

In addition, the creator of the Knowledge Type has control over who can modify and delete the Knowledge Type. Each Knowledge Type maps onto a set of group-based Knowledge Type permissions, specifying the groups that can modify, delete and use this type.

The permissions specified for a Knowledge Type apply to all Knowledge Fields of that type, thus making it impossible to give one group of users permission to modify the title of an article and a different group of users to modify its abstract. The latter was a consciously made design decision: specifying separate permissions for each part of a document would complicate browsing and editing and does not seem to be an especially useful capability.

5.3.2 Knowledge Object Authorization

Knowledge Objects have permission attributes that are distinct from those of their Knowledge Type. Only those users who have access to a knowledge type can create, modify and view objects of this type. In order to make the authorization model more intuitive, the structure of setting permissions for a specific object is similar to the one for a knowledge type -- the user specifies the groups he/she wants to be able to modify, delete, and use (in other words, view) the Object. Such fine-grained permissions help provide users with flexibility in managing content, making it easier to share information without giving up control or endangering valuable resources.

The system also provides a simple version control mechanism for modifying/deleting objects. Only a single user can modify the Object at a time. Each Object has a Status attribute - it is either marked as Checked In or Checked out. The Status attribute represents whether the Object is currently being modified. If the Object is Checked Out, the identity of the user who currently has control over it is also kept track of. In the future, a more elaborate version control might be desired, such as keeping track of old versions and allowing multiple users to modify an object and then merging the changes.

5.4 Search

For a Knowledge Management system to provide for efficient navigation and filtering of content, it has to offer search capabilities. The Know-and-Share Center currently features a simple Search utility. A user can enter one or more words and the search will return a list of Objects that contain any of those words in any of their Knowledge Fields, included in a site-wide search by the type creator, or Comments.

Each Object has a system-defined Keywords Field. If that field contains any information, the Find Similar Objects tool becomes available to all users who can view the Object. The tool performs a search for words contained in the Keywords field and returns a list of Objects containing one or more of those Keywords.

In the future, the system may benefit from a more sophisticated Search tool.

5.5 Personalization

This section describes those features of the system that makes the experience of a single user more enjoyable. Since the system is capable of handling a great number of knowledge types and objects, a single user may be overwhelmed with the amount of information he is presented with. Thus, a few additional features necessary in order for him to organize his work space.

Bookmarks

While the users certainly benefit from having a big collection of Knowledge Objects as a resource, most users are working with a relatively small subset of objects at a particular point in time. Thus, it is important to provide a mechanism to allow the user to take an object and drop it into a personal folder so that, when the user needs the object again, it is within one click of the mouse.

The Bookmarks feature of the Know-and-Share center provides the capability described above –– it lets each user keep a personal list of frequently used objects. It makes sense for the Bookmark list to be distinct for each user –– user's interests and preferences are certain to vary widely. Offering the user to create folders and drop objects into these folders (similar to the bookmarking feature in Web browsers) was considered and decided against in order to simplify the bookmarking mechanism and not overwhelm the user with redundant functionality.

Preferred/Non-Preferred Types

A user will rarely find all Knowledge Types available to him to be equally interesting. The division of Knowledge Types into Preferred and non-Preferred is a personalization tool that allows the user to mark some types as less interesting. Objects of the user's Non-Preferred Types will always be listed below Objects of the user's Preferred Types. Thus, the Preferred/Non-Preferred feature allows the system to take into account the user's interests when presenting information. The user can change a Knowledge

Type's status from Preferred to Non-Preferred or back at any time. Each user has his/her own Preferred/Non-Preferred settings, reflecting the fact that every user's interests and preferences are different.

5.6 Object Links

An important aspect of a knowledge management environment is that an object is rarely completely detached from all others: most of the time, the system contains multiple related documents that may provide useful additional information on the topic the user is interested in. A feature offering users a capability to link Objects is thus an important factor in supporting effective sharing and easier navigation of knowledge. The Object Links utility provides a mechanism for creating navigation threads –– paths that will take a user along a thread of related documents. A user can create a link between any two objects, as long as he/she has permissions to view both of them.

Links are uni-directional: thus linking Object A to Object B is not the same as linking Object B to Object A. This design decision is consistent with the fact that the connection between two Objects is often anti-symmetrical. For example, if an article mentions that today's young programmers need a role model and a user reading the article adds a link to Bill Gates' bio, immediately creating a link back would make very little sense. In the future, however, an option to provide a bi-directional link might be desired.

Links can be Public (only visible to the author) or Private (visible to anyone who has the right to view both Objects). The user can attach a comment to a link. The comment is useful as a reminder to the author and an explanation to others of why the link was created and how the Objects are connected. Implementing more elaborate authorization for viewing links (e.g., only certain groups see a link) would have added an unnecessary level of complexity and was rejected in favor of a more simple model in order to make the use of the system and creation of links as effortless as possible.

5.7 Comments

The Know-and-Share Center provides users with a way to comment on a particular field of a particular Knowledge Object. The fact that the user has the ability to comment on a particular field of an Object helps make the comments more specific. A user may choose to designate a Comment as private, thus making it invisible to all other users. He/she may also designate the Comment as Public, making it available to users who have viewing permissions for the object. A public comment may be

Signed (users who read the comment will also see its author's name) or Anonymous (unsigned).

It is currently impossible to create a comment that is available to a limited group of users –– each comment is either Private and only visible to its author or Public and visible to anyone who can view the Object. This design decision was motivated by the fact that assigning permissions to a comment would require the author to go through an additional screen and fill out an additional form. The goal of the Know-and-Share Center is to promote collaborative development of knowledge and encouraging users to comment on Knowledge Objects helps achieve that goal. Thus, the process of creating a comment should be as quick and easy as possible.

The Comments feature of the Know-and-Share Center serves three important purposes:

1. Public comments serve as a filtering tool. They enable users to benefit from other's judgment and thus read a richer mix of material and save time that would otherwise be spent skimming and rejecting.
2. Public comments can also be used as a tool for content criticism. Criticism is crucial to ensure continuous evolvement and development of high quality content. Also, the concept of managing knowledge online offers a unique opportunity for supporting easier and faster criticism:
   - Critics can be more effective for less effort. They don't have to summarize the content of the document –– the reader will see the criticism while he/she is going through the document.
   - Ease of submitting criticism will make authors more careful about what they post –– users will be more reluctant to create Knowledge Objects of doubtful quality.
   - Users can post criticism on criticism. If a particular comments seems malicious and ill-founded, users can comment on the comment.
3. Private comments can be used the same way as one would use notes on the margins of a book. When a user revisits a particular Knowledge Object, he/she will be able to the comments he/she made earlier.

5.8 Ranking

While one of the goals of the system is to accumulate high quality content, the Knowledge available to the users of Know-and-Share will certainly vary in usability. Users will like some Objects better than others and it is desirable to provide them with a tool to express their opinions. By letting users evaluate the system content, the Ranking tool helps the system achieve two of the objectives defined in Chapter 3:

- promoting product quality – the way users rank knowledge lets the creators of the knowledge infer the level of quality of particular Objects
- promoting personalization – ranking helps users filter the body of knowledge Know-and-Share offers, providing for more effective use of the system

The Ranking choices are currently "Liked It", "It's OK", and "Didn't Like It". The choices are represented by entries in a system table and can thus be added or modified easily if the need arises.

The information about the ranks that different users assign to an Object is used to sort Object lists users are presented with. For example, when a user views a list of Objects of a particular Type, he/she can choose to sort the list using the following metrics:

- alphabetically –– useful when the user is trying to locate an object by name
- by personal rank –– Objects the user gave a positive ranking ("Liked it") are listed first, followed by Objects with a neutral ranking ("It's OK"), Objects the user has not ranked yet and finally Objects with a negative ranking ("Didn't like it")
- by average rank –– works similarly to the personal rank sort, except that the average of ranks given by all Know-and-Share users to each Object are used instead of personal rank

Ranking is a mechanism for improved filtering of information. It lets the user build up a subset of the knowledge base he/she finds most useful, thus making Know-and-Share a more valuable resource.

Chapter 6

Implementation

This chapter presents an overview of the implementation of the Bridge Group Community environment. First, it describes the tools that were used in creating the system, in particular the ArsDigita Community System (ACS) and the modules of ACS that were heavily used in the system implementation. Then, a short overview of the user interface to the system is given. Later sections describe in detail the data model and implementation chosen for the core components of the Know-and-Share Center: Knowledge Types, their corresponding Type Fields and Knowledge Objects, which were mostly implemented by the author of this thesis. Finally, the chapter gives a brief overview of the implementation of other parts of the system, such as authorization architecture, user groups, bookmarks, links, search, ranking and comments, which were primarily implemented by my teammates, Lyudmila Zemlyakova and Sofya Pogreb. The data model for the knowledge center, permissions and survey modules are included in Appendix A.

6.1 ArsDigita Community System

The Bridge Group Community system has been built using the ArsDigita Community System (ACS) and an AOLServer. Rather than developing a knowledge management system from scratch, the ACS was chosen for its variety of modules that provide extensive functionality such as discussion forums, chat rooms, and user-group data model. The ACS makes a heavy use of Tcl as the programming language, AOLServer as the web server, and Oracle database as the information storage, thus these were also used in developing the system. This choice of technical tools best provided the functionality required for the system. The system uses the Users and User Groups modules of ACS. In order to provide the look and feel of a close community, several ACS modules (such as chat and discussion board) needed to be integrated into the system. A new Survey module was added to the system in order to provide an improved way of gathering user feedback (previously, a simple poll module with very limited capabilities was used for such purposes). This module was requested by the Bridge Group administrators that expressed their desire to use it primarily for collecting feedback from the website users about the system as a whole, as well as for obtaining information about the non-profit world.

6.2 User Interface

The major part of the site navigation strategy was to allow users to go to any major module of the Bridge Group site with just one or two clicks. This decision led to the idea of the BridgeBrowse bar which would be present on every page of the site. This bar contains links to the following sections: the home page, the information about the Bridge Group, consulting services that the Bridge Group offers, the Know-and-Share center, client groups management, feedback to the Bridge Group, contact information, and site map. The navigation of the site itself and especially the Know-and-Share Center is straightforward and has a uniform look throughout the site. As many options as possible are presented on a single page without overwhelming the user with irrelevant information. Additionally, the information presented on the homepage can be modified by administrators of the site through the Know-and-Share center. Thus, a non-technical person can, for example, easily change the "What's New" section without changing the underlying code or HTML for this page. This part of the system was implemented primarily by Lyudmila Zemlyakova.

6.3 User Groups and Users

The User Groups ACS module is the bulk of the Client Group section of the site. The ArsDigita software has a powerful user group module, which was modified and used in this part of the system instead of developing the user group support from scratch. There are two major user group types that the users can extend: the Bridge Group Client Organizations and the Knowledge Center Users. The Bridge Group Client Organizations type will include all client organizations of the Bridge Group and the group creation requires an administrator's approval (as was described in the design chapter). The Knowledge Center Users type allows users to create their own private groups to manage knowledge types and objects, which do not require approval. More user group types with various attributes can be added later by an administrator.

The ACS module provides a good support system for registering new users and monitoring actions of registered users. In order to provide the necessary level of security for the content created by the users, non-registered users of the site do not have access to the Know-and-Share module. Users that don't belong to any of the Bridge Group Client Organizations (the top level groups) can only view the content that is available to all Website users. Registered users who belong to one or more Client Organization can not only view content created by other users in the system (if they are authorized), but create their own user groups and create content which can have various levels of security.

6.4 Administration

There are three abstract types of administration in the system, which are based on the user groups and user roles in those groups: the Site Wide Administration, which mainly consists of the Bridge Group employees and site developers; the User Group Administration, most features of which are available to all users without restrictions (except for creation and management of top level client groups, which require an approval of a Site Wide Administrator), and the Know-and-Share Administration which provides a comprehensive set of features for creating and sharing knowledge, which is once again available to all members of client groups.

6.5 Knowledge Types

The Know-and-Share system gives great flexibility in choosing the structure for a new knowledge type and provides the ability to modify existing knowledge types. The design of this part of the system focused on making the users' job easier and provide comprehensive mechanism to manage knowledge. When a user decides to create a new object type, a Tcl page is presented to him/her with a few fields that he/she needs to fill out, which include a short name (which will become part of the name for a dynamic table that will contain objects of this newly created type). He/she also provides a name for the knowledge type, which will be displayed to the users creating objects of this type. The creator of the new knowledge type is also prompted to set permissions for this type, which will be described later in this chapter.

In order to represent different types of knowledge in the system (e.g., articles, cases, funding proposals, etc.), the system keeps an SQL table called km_metadata_object_types. When an knowledge type is created, a row is added to the km_metadata_object_types table. This table contains information common to all object types. In order to provide extensive flexibility for the users creating new knowledge types, only a few attributes are common across all types, such as names and the author.

After creating a knowledge type, the user can modify any attributes at any time, except for the short name since it's part of the dynamic object table name, and it requires a great deal of effort to alter the table name in Oracle.

Along with adding a row to the knowledge type table, a new dynamic table is created, which will contain objects of this type with a few default columns. The following is an example of an object table, which is created for a knowledge type "Article":

```
create table km_article (
        km_object_id            not null references km_all_objects,
        name                    varchar(200) not null,
        overview                varchar(4000),
        created_by              not null references users,
        date_created            date default sysdate not null,
        modified_by             references users,
        date_modified           date,
        include_in_ctx_index_p  char(1) check (include_in_ctx_index_p in ('t','f')),
        keywords                varchar(4000)
)
```

The columns in the dynamic table correspond to the default attributes of an object and the knowledge fields that are added to the knowledge type by the knowledge type creator.

The next step in defining the structure for a knowledge type is to add customized fields that will differentiate it from other types. The mechanism for doing that is described below.

6.6 Knowledge Type Fields

When the user creates a field, he/she associates it with a particular knowledge type. For example, a field called "Published by" can be associated with a knowledge type called "article". One of the main goals of the Know-and-Share system is to achieve the dynamic manipulation of knowledge types and objects. This section discusses how the dynamic features of the Knowledge Center have been achieved through manipulation and altering of Oracle tables to support dynamic changes to the knowledge types.

At this point, the system allows a many-to-one mapping between fields and knowledge types. In the future, however, it might be desirable to associate fields with multiple knowledge types to promote reuse of the fields.

The user pages facilitating creation and modification of knowledge type fields were designed to walk the user through the process, thus making it as technically undemanding as possible. It requires the user to fill out only two short forms, which will lead to creation/modification of the field. As described in the scenario, the first form prompts the user for the name of the field and the abstract data type, which can be one of the following:

- Text (less than 4000 characters)
- Long Text (more than 4000 characters)
- Character
- Number
- Date
- Selection (checkbox, radio and select HTML tags)
- File

These data types represent the type of input that is required of the user filling out this field when he/she is creating or modifying knowledge objects. The abstract data types are used to filter out those HTML input types that cannot be used to represent information that needs to be captured by this knowledge type field. For example, the data of type "Long Text" can only be entered in an HTML "textarea" input field. Thus, the second form of the field creation/modification process depends upon the abstract type that he/she chose in the first form. It asks the user to select an HTML input type, which will be used to present this field to a person creating a new object. This mechanism requires the user to be familiar with how different form attributes work in HTML since creation and modification of objects is done on the Web through HTML forms. Some knowledge management systems that exist in the market today use a different approach -- they ask the user for some abstract description and attributes of the knowledge type and fields (e.g., graphically select the way to represent a knowledge field), but they usually require special software to prompt the user for such input and cannot create knowledge fields through the Web. Although the decision to use HTML forms and abstract types has a few disadvantages, such as the limited ability to represent data and the requirement of the user to be familiar with some HTML, our design choice is best suited for online modification of knowledge types.

The following mapping was used to eliminate some of the HTML input types depending on the user-selected abstract type:

- HTML "Text Field" input is used for the Character, Number and Date data types.

- HTML "Text Area" input is used for the Long Text data type.
- HTML "File" input is used for the File data type.
- The user has a choice between "Text Field" and "Text Area" input types for the Text data type.
- The user has a choice among "Checkbox", "Radio", and "Select" HTML input types for the Selection abstract type.

Additionally, the user is asked to indicate whether he/she wants the field to be mandatory, included in the knowledge center search, etc. as described in the scenarios.

After all the information is obtained from the user, a row is added to the "km_metadata_elements" table. The columns in this table capture both the information necessary to present this field to the user (such as abstract_data_type, presentation_type, presentation_options, form_sort_key, etc.) and the information used by Oracle to alter the dynamic table used to store objects of this knowledge type (such as oracle_data_type and extra_sql). Most information is directly stored in the table, except for the oracle_data_type, which is inferred from the abstract data type chosen by the user. The following inference rules are used (in the future, a more fine-grained mapping might be desirable to save space in the Oracle database).

- Text (less than 4000 characters) maps to "varchar(4000)" Oracle data type
- Long Text (more than 4000 characters) maps to "clob()" Oracle data type
- Character maps to "char(1)" Oracle data type
- Number maps to "number" Oracle data type
- Date maps to "date" Oracle data type
- Selection (checkbox, radio and select HTML tags) maps to "varchar(4000)"
- File maps to "varchar(4000)" Oracle data type

The next step is to modify the dynamic object table associated with the knowledge type being modified. A column is added to the table in the following format:

```
alter table km_article
        add (published_by varchar(4000) default 'New York Times')
```

Once the field has been created, some of its attributes can be modified. For simplicity of implementation, an authorized user can only modify the presentation options of a knowledge type field, not the Oracle attributes. This choice was made to prevent loss of data. For example, if a few articles have been created by users, and the the data type of a certain field has inadvertently changed (e.g., from clob() to varchar(200)), some data may be lost. In the future, additional capabilities, such as changing the Oracle data type can be implemented and supplemented by mechanisms to prevent the loss of data.

6.7 Knowledge Objects

As was mentioned above, tables that contain information about knowledge objects are generated dynamically when a new knowledge type is introduced. New columns are inserted into the table when fields are added to the knowledge type. Analogously, the columns are deleted when these fields are removed. Currently, the system allows authorized users to delete fields from the knowledge type, which might result in a significant loss of data. Certain care should be taken with respect to setting permissions. Our authorization model and reasons behind it are described later in this chapter.

The mechanism to dynamically alter the table (through inserting and deleting columns) was designed to allow users to modify the knowledge types after objects of this type have been created. An alternative would be to ask the user to create a knowledge type with desired fields and then run a script to generate a "static" object table with columns corresponding to the knowledge fields. However, this would prevent administrators of knowledge types from modifying the type after it has been created. In many systems, this would be a desired choice when administrators of the system carefully consider the types of knowledge needed, and the changes to those types occur very rarely. However, the Know-and-Share system encourages not only the evolution of knowledge objects and collaboration among users on adding more information to the system, but the evolution of the knowledge types themselves.

6.7.1 Creating, Modifying and Deleting Objects

A knowledge object can be created by any user authorized to use its corresponding knowledge type. By following the "Create New Object" hyperlink from the corresponding knowledge type, the user is presented with a form, which he/she has to fill out. An example of such form is shown by Figure 2-12 and the process of creating an object is described in the Scenarios chapter.

The form itself is generated dynamically given a knowledge type. There are two default visible attributes that all the knowledge system objects have to have (they are not part of the metadata fields added by the users): "name" and "overview". Moreover, the "name" attribute is mandatory, and an object is not created until it's given a name, while an overview section is optional. As shown in section "Knowledge Types", object tables contain some additional attributes, which are not visible to the user who is creating or modifying the object, but are used by the system mostly for checking permissioning. In the future work, these attributes can be used for reporting and monitoring purposes.

Each column in the object table, which is part of the km_metadata_elements table, is displayed according to its presentation type and options. This and some additional information, such as the name of the field and its default value is obtained mostly from the km_metadata_elements. The creator of the object is also prompted to fill out all mandatory metadata fields, which is checked by the system during the submission of the form. At this point, the user also sets permissions for the object. It takes only two HTML forms to create an object. Once he/she submits the newly created object, it is immediately available for viewing.

6.7.2 Displaying Objects

The process of displaying a knowledge object is similar to its creation. Each column in the table is analyzed and presented to the user given the information in the km_metadata_elements and the dynamic object table itself. Again, careful permission checking is done before the person can view the content.

6.8 Permissions and Security

A user can give permissions to the following types of user groups and users: himself, other individual users (he specifies their email addresses), to All Website Users, to any group he/she is a member of, and to any Client Organization group (which are top level groups in the system). Allowing the user to share his/her content with any group in the system would overwhelm the user with the number of different groups in the system and prevent him/her from carefully setting permissions for the content, thus he/she has a limited choice of groups, which he/she can authorize to modify or use the content. Note, however, that additional single users can be authorized as well by entering their email addresses in the appropriate fields when setting permissions.

A person creating a knowledge type gives permissions to modify, delete, or use this knowledge type and its fields. The permissioning model for individual knowledge objects is similar to the model for knowledge types with the addition of the object viewing permissions.

Since the system is intended to handle multiple user accesses to the same object, problems may arise when more than one user try to modify the same object. To prevent this, the system provides a mechanism to "check out" an object when it is being modified. The user who chooses to modify the object locks it, and the system does not allow anybody else to modify it at the same time. Authorized users can still view the old version and see the changes made to the object only after the user who has made those changes checks the object back into the system.

6.9 Personalization

The following subsections briefly describe the implementation of features that facilitate personalization and navigation in the system. This work was done by Lyudmila Zemlyakova and Sofya Pogreb.

6.9.1 Comments

A comment link is provided for each section of an object. A comment link next to the name of an object is designed to comment on an entire document. When the user clicks on a comment, he/she can review or edit his comment, add a comment if it does not exist or view other people's public comments. The comment link appears red if the section is commented (either by the user himself or by other people), which frees the user from unnecessary browsing. Comments can be private or public.

6.9.2 Object Ranking

Object Ranking is a tool that provides for user evaluation of the content. When the user views an Object, he/she is offered to rank it. Once the object has been ranked, the ranking can always be modified.

Ranks submitted by the user are utilized to personalize the Object lists that are presented to him/her. The average of ranks that all Know-and-Share users assigned to a particular object can also be used to sort Object listings and reflects the quality and usability of the Object as seen by the community of users.

6.9.3 Object Links

Links are implemented as a table that maps Knowledge Object A to Knowledge Object B. A link can be created by any user who has permissions to view both Objects. Each link includes a comment in which the user can explain why he/she feels the objects are related and therefore should be connected. Additionally, the user can make these links be available to the public.

6.9.4 Bookmarks

Bookmarks serve to facilitate knowledge use by letting the user get to his/her favorite objects with a single mouse click on a link located on the Bridge Browse Bar. Bookmarks are stored in a simple table that maps a User ID to an Object, representing the one-to-many user-object relationship. There is no limit to the number of objects a user can bookmark.

6.9.5 Preferred/Non-preferred Object Type

The Preferred/Non-Preferred Types tool allows the user to mark selected Types as less interesting. The resulting effect is that the objects of Non-Preferred Types are listed below those of Preferred Types when the user views Objects available to him/her.

6.9.6 Know-and-Share Search

The Know-and-Share Search feature is implemented as a simple keyword search. Search results are compliant with the user's access privileges. They can be sorted alphabetically or by their rankings. In the future, a more advanced search would be desired.

The "Find Similar Objects" feature appears on the page along with the content of a Knowledge Object. By clicking on on it, the user is presented with similar objects in the system. This feature is implemented directly using the Know-and-Share Search tool.

Chapter 7

Experiment

The application described in this paper has been turned over to the Bridge Group. The Bridge Group staff is now working out certain administrative details that have to be resolved prior to actual deployment. Thus, we were not able to test the Bridge Group Community System with a realistically large population of users. However, to test how well the Bridge Group Community System achieves the goals we posed in the design section of this document, we asked the Bridge Group employees to devote some time to using the application over a period of two weeks. Our test group came back to us with extensive feedback and suggestions, which we discussed with them and documented.

The Bridge Group staff was very pleased with the system we presented to them -- they appreciated the extensive capabilities offered to the users as well as the small time commitment required to administer and maintain the system. It has to be noted, however, that, until the system is used by a large number of organizations, the administrators' load will not reach its maximum. Test users created several new Knowledge Types, but, in most cases, the Knowledge Types were reused. Some of the concerns the test group brought up were:

- the administration pages aren't easy enough to navigate and contain a lot of unnecessary capabilities that are very unlikely to be used
- the classification of knowledge by Knowledge Type is not convenient from the user's point of view
- while the Bridge Group wanted to minimize the site administrators' workload, the management was concerned about being able to trust users previously unfamiliar with the environment to create arbitrary new knowledge types
- the chat tool and the discussion forum mechanism aren't well integrated with the Know-and-Share Center

- a new user has to wait for his/her membership to be approved before being able to take full advantage of the Know-and-Share center

We have implemented a number of features and improvements suggested to us by our test users as a result of the concerns listed above. The reasons we felt these alterations were desirable, as well as the implementation choices we made, are presented below. Other suggestions were too implementation-intensive given our time constraints and are described in the Future Work section of this document.

7.1 Making New Users Wait to be Approved

The fact that a new user has to be approved before being able to use all features of the Knowledge Center bothered the Bridge Group staff because a potentially valuable contributor may be driven away by inability to immediately see what the system has to offer. A possible solution is allowing a new user full access to the knowledge immediately after registration. However, the Bridge Group was unwilling to accept the latter option. The management suggested that a system-wide password can be given out to client organizations, so that any new user who works for a client organization should be able to provide the password and be immediately approved by the system. The suggestion was considered and rejected because having a system-wide password is an unacceptable security hazard – people who have worked for a client organization in the past will know the password, thus having access to information that is no longer relevant to them.

7.2 Classification By Subject

One of the strongest criticisms expressed by the test user group was that classification of Knowledge according to Knowledge Type was not sufficient. While a Knowledge Type describes the structure of a Knowledge Object, it does not say anything about the topics the Object relates to. Thus, the test group suggested that a classification by Subject should be added to the system.

It is hard to disagree with the latter criticism – indeed, classification of knowledge by its structure rather than by topic is oriented towards knowledge authors rather than knowledge users. The Bridge Group Community system aims to promote both the creation and the usage of knowledge. Thus, a classification by subject might be valuable for those who want to use the Know-and-Share.

To implement the classification by Subject feature, a new table was added to the database to represent the mapping from an Object to a Subject. When a user lists the Knowledge Objects, he/she has an option to group them by Knowledge Type or by Subject. Thus, someone who is looking for an article may prefer listing by Type, while someone trying to find information on fundraising would probably choose to group the knowledge by Subject.

A mechanism to allow every user to add a new Subject has been put in place. However, the Bridge Group staff expressed a desire to restrict the creation of new Subjects to the community of Bridge Group employees. Thus, when Bridge Group employees go to the KM Administration page, they are presented with a list of currently existing Subjects, as well as an option to create a new Subject.

When a user creates a new Knowledge Object, he/she is presented with a multiple select box allowing him/her to select one or more Subject to associate the Object with. If the user feels none of the Subject are related to the Object being created, the field can be left blank and the Object will be automatically listed under the Subject "Other".

7.3 Search By Subject

A natural extension of the classification of Knowledge Objects by Subject is a feature allowing the user to perform a search on Objects that belong to one or more particular Subjects. On the search form, besides entering keywords to search for, the user now has an option to select subjects to conduct the search among. Alternatively, he/she may choose to search all subjects (please, refer to the search page and the results page in Figures 7-1 and 7-2).

While the search tool remains quite simple, offering the users a capability to search by subject makes it much easier to find knowledge dealing with a particular topic of interest.

7.4 Reorganization of the Administrator Pages

One of the most important requirements of the Bridge Group has been minimizing administration efforts. However, the ACS Software administration pages are not straightforward and easy to navigate for users with little technical background. Moreover, the system provided many more administrative capabilities that the Bridge Group administrator will ever need. For example, the Bridge Group site currently has only three modules from the ACS software, that are visible to users (those were the modules specifically requested for the site by the Bridge Group). They are the discussion forum, chat, and the survey (implemented by us based on the idea of polls in ACS). Also, the Bridge Group administrator wants to view the users and user groups and send emails to them if necessary. Finally, he/she needs to maintain static content on the main page of the site (adding alerts and "what's new" features). Figures 7-3 and 7-4 at the end of this chapter illustrate how the index page for administrators looked before and how it looks after we removed unnecessary modules and grouped the remaining features.

Also, we had to substantially change the navigation process throughout the administrative pages and add more help comments. As a result of our changes, the administrative pages have become much easier to navigate. Thus, with the initial ACS administrative pages, there was a learning curve for an administrator in order to effectively navigate and administer the site. Many questions have been posed by the Bridge Group employees to us about the administrative features and tools. Presently, there are many more explanations and comments on the pages, unnecessary tools and features have been removed, and thus, the learning curve of a new administrator has been substantially reduced.

Another feature requested by the Bridge Group was to add a notification mechanism for users who have expressed desire to be notified when new features and information are added to the system. Thus, a Bridge Group administrator can now spam these users from the administrative pages whenever he/she wants to notify them of new developments on the site.

7.5 Approval of New Knowledge Types

The test group felt that, while letting community members create new Knowledge Types is a great capability that helps promote creation of new knowledge, an average users cannot be trusted to create a well-designed Knowledge Type. The Bridge Group wanted, at least in the initial phase of the system's existence, to have some control over the Knowledge Type creation process. Thus, it was suggested to us that every new Knowledge Type should be subject to approval by a member of the Bridge Group staff.

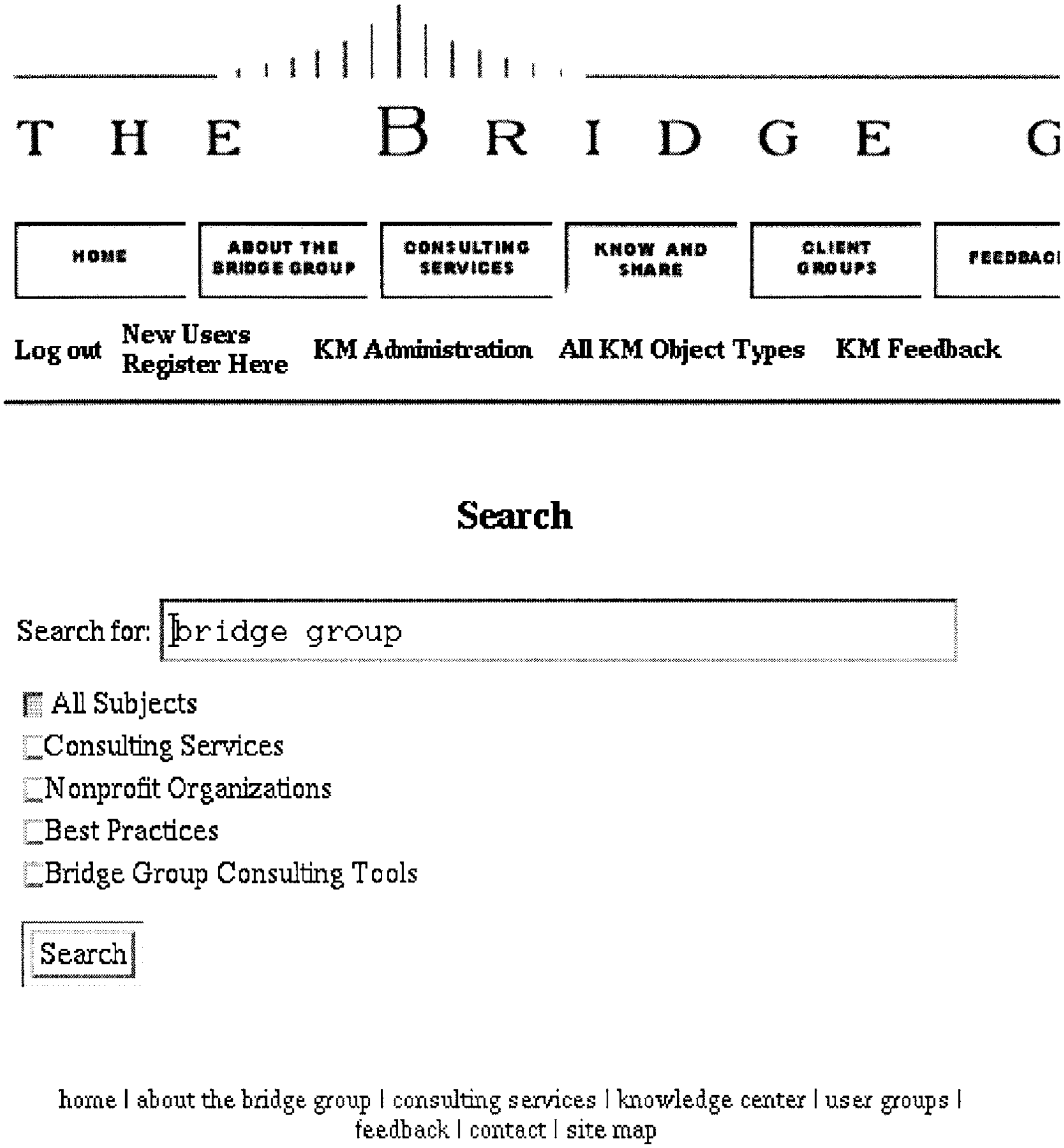
The suggestion certainly makes sense and the Bridge Group's desire for more control (at least initially) is very understandable. Thus, users still have the capability of creating a new Knowledge Type and specifying Knowledge Fields that will make up the Knowledge Type. When the Knowledge Type is created, the user receives an email informing him/her that the Knowledge Type has been submitted for approval by the Bridge Group and that, until it is approved, Objects of that Type cannot be created.

When a member of the Bridge Group staff logs in and visits the KM Administration page, he/she will be presented with a list of types that are awaiting approval. The Bridge Group employee can do one of the following:

- approve the Knowledge Type – the author will receive an email informing him/her the Type has been approved and the Type will become available for use by the members of the community specified by the Type's permission settings
- refuse the Knowledge Type – the author will receive an email informing him/her the Type has been refused and a list of reasons for the decision; the Type will be deleted from the system
- email the author asking him/her to edit the Type definition in a certain way and resubmit it for approval
- do nothing and leave it up to other Bridge Group staff members to decide to approve/refuse the new Type

It was decided not to make the new Type available for use until it has been approved. The latter decision presents a tradeoff. On one hand, a user created the type because he/she wanted to author a Knowledge Object which didn't fit well into any of the existing Type definitions. Thus, making the user wait for an approval delays him/her from authoring the Knowledge Object right away, potentially discouraging creation of new knowledge. On the other hand, however, allowing creation of Objects of the new Type before type approval presents the following dilemma: if the Bridge Group staff does not feel the Type is needed, the knowledge object will have to be deleted. We felt that making the author wait is preferable to losing knowledge – thus, it was decided not to make the Type available for use until it has been approved.

Overall, the experiment has shown that the design of the system satisfies the client's major requirements – extensibility and ease of administration. However, the Bridge Group is not comfortable with letting new users exploit all capabilities the system originally allowed for. The test group requested that, initially, newly created Knowledge Types should require approval. The latter example shows that, although the Bridge Group desires extensibility, completely giving up control is a price the company is not willing to pay. As a consequence, checking mechanisms, such as requiring approval, need to be put in place, thus increasing the administrators' load. Hopefully, in the future, as the users become more accustomed to the system, the Bridge Group staff will be able to trust them more, and the time needed to administer the system will be reduced.

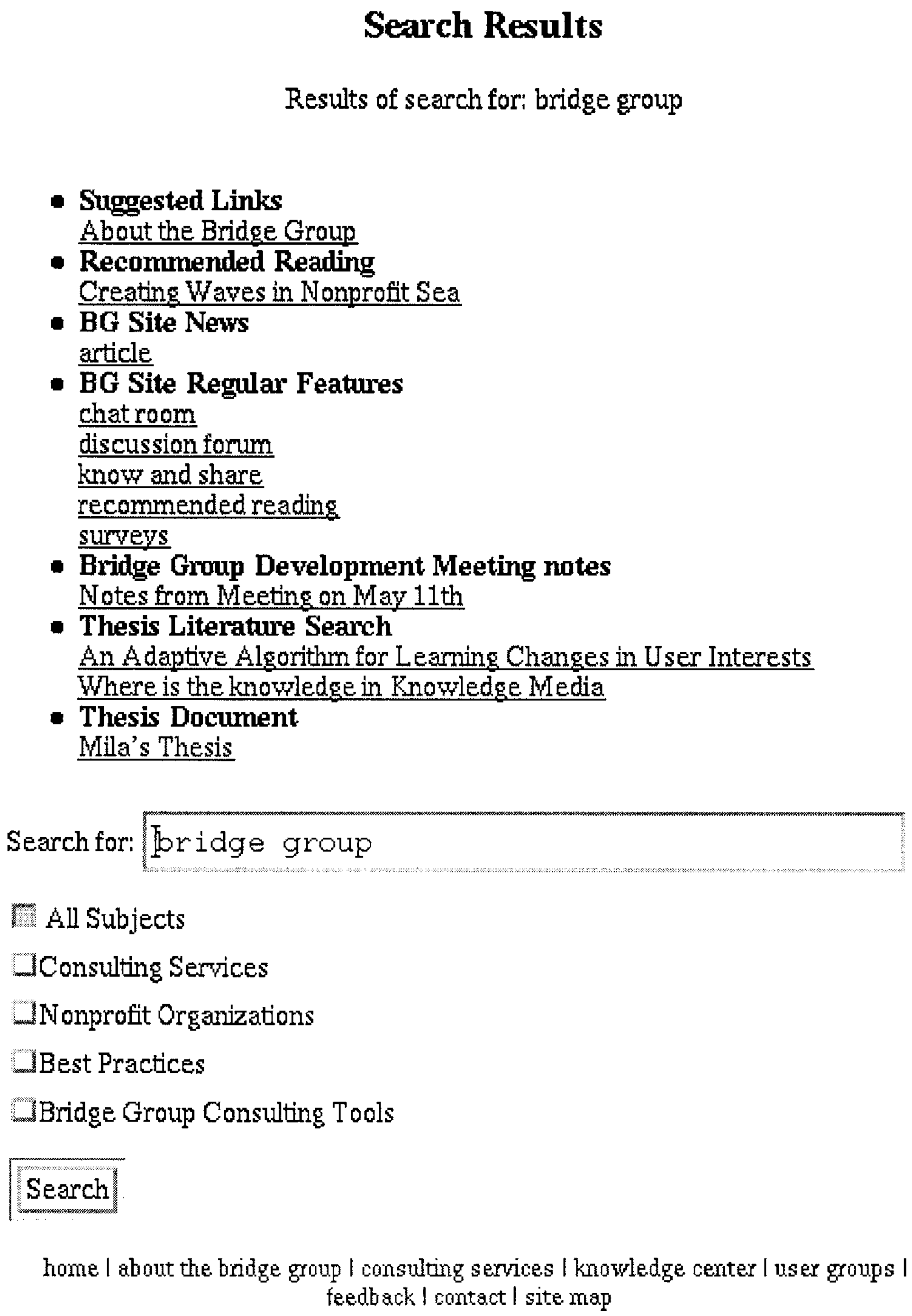


Search

Search for: bridge group

☒ All Subjects
□ Consulting Services
□ Nonprofit Organizations
□ Best Practices
□ Bridge Group Consulting Tools

Figure 7-1: Search by Subject Page for Know-and-Share



Search Results

Results of search for: bridge group

- **Suggested Links**
  About the Bridge Group
- **Recommended Reading**
  Creating Waves in Nonprofit Sea
- **BG Site News**
  article
- **BG Site Regular Features**
  chat room
  discussion forum
  know and share
  recommended reading
  surveys
- **Bridge Group Development Meeting notes**
  Notes from Meeting on May 11th
- **Thesis Literature Search**
  An Adaptive Algorithm for Learning Changes in User Interests
  Where is the knowledge in Knowledge Media
- **Thesis Document**
  Mila's Thesis

Search for: bridge group

☑ All Subjects

□ Consulting Services

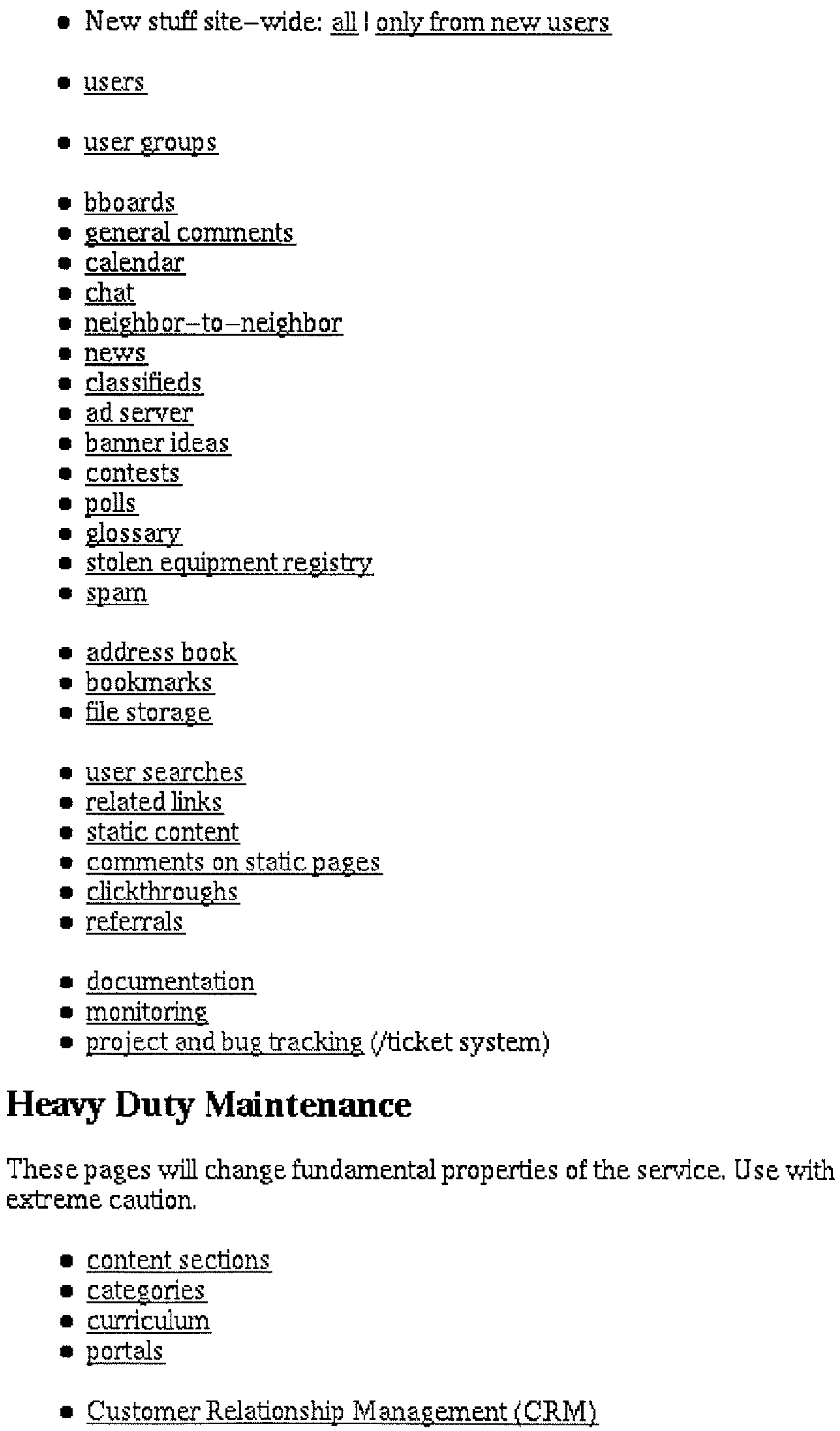
□ Nonprofit Organizations

□ Best Practices

□ Bridge Group Consulting Tools

home | about the bridge group | consulting services | knowledge center | user groups | feedback | contact | site map

Figure 7-2: Results of Search by Subject Page for Know-and-Share



- New stuff site–wide: all | only from new users

- users

- user groups

- bboards
- general comments
- calendar
- chat
- neighbor–to–neighbor
- news
- classifieds
- ad server
- banner ideas
- contests
- polls
- glossary
- stolen equipment registry
- spam

- address book
- bookmarks
- file storage

- user searches
- related links
- static content
- comments on static pages
- clickthroughs
- referrals

- documentation
- monitoring
- project and bug tracking (/ticket system)

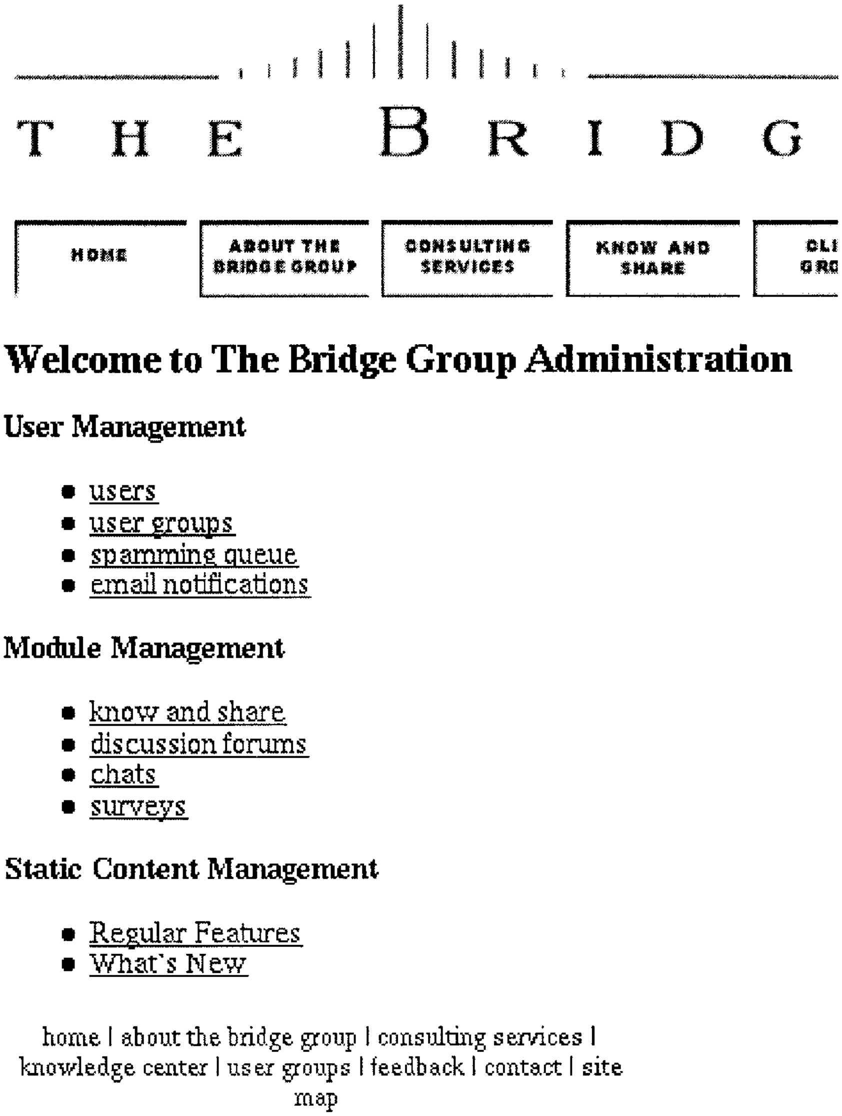
Heavy Duty Maintenance

These pages will change fundamental properties of the service. Use with extreme caution.

- content sections
- categories
- curriculum
- portals

- Customer Relationship Management (CRM)

Figure 7-3: Old Index Page for the Bridge Group Administrator



Welcome to The Bridge Group Administration

User Management

- users
- user groups
- spamming queue
- email notifications

Module Management

- know and share
- discussion forums
- chats
- surveys

Static Content Management

- Regular Features
- What's New

home | about the bridge group | consulting services | knowledge center | user groups | feedback | contact | site map

Figure 7-4: New Index Page for the Bridge Group Administrator

Chapter 8

Competitive Analysis and Evaluation of Accomplishments

This chapter compares the Bridge Group Community System to the current state-of-the-art Knowledge Management Environments and evaluates the extent to which the Bridge Group Community achieves the goals stated in the Goals section of this paper.

8.1 Functionality to Support Effective Knowledge Management

It is easily noticeable that the systems analyzed in the related work section of this paper differ greatly in functionality and number of features. It would have been easy to compare those systems by their features and conclude that the one offering greatest functionality must be the best. However, although extra functionality usually (not always) improves the users' experience, it is important to keep track of how well the features are chosen, designed, and used to achieve more fundamental goals.

The Related Work Section of this paper describes the functionality a Knowledge Management Environment should aim to support. The four tables below serve to evaluate the extent to which the Bridge Group Community System and other systems being analyzed achieve the following four fundamental goals:

1. Promoting Creation of Knowledge

2. Promoting Evolution of Knowledge

3. Promoting Sharing of Knowledge
4. Promoting Use of Knowledge

The tables presented in the next few pages address those four issues and show whether and how well each of the four systems (plus the Bridge Group Community system) described in the related work section address these issues. The rows of the tables represent features that help achieve the goal. Each column of the tables refers to one of the systems being evaluated. The column numbers refer to the systems as follows:

1. Bridge Group Community
2. Intranets.com
3. Novation
4. SiteStation
5. Xpedio

1. Promoting Creation of Knowledge. A successful knowledge development system should provide an efficient mechanism for creating knowledge. The criteria chosen to evaluate different systems with respect to this goal mostly include features that allow different users to contribute to the content created in the system, not only a handful of administrators. It should be easy for regular users to publish their ideas and share them with the rest of the online community. In Table 7.1, "users" refer to the majority of people using the system, not only the designated content originators and publishers.

Promoting Creation of Knowledge	**1**	**2**	**3**	**4**	**5**
Users can contribute to knowledge base	Y	Y	N	Y	N
Mechanism exists for directly submitting content	Y	Y	N	Y	N
Mechanism exists for imposing some structure on the knowledge types	Y	N	Y	N	Y
Users can add new knowledge types	Y	N	N	N	N
Mechanism to create knowledge types using existing types	N	N	Y	N	N
Users can edit the structure of existing types	Y	N	N	N	N

Table 8.1: Promoting Creation of Knowledge

2. Promoting Evolution of Knowledge. This goal deals with the ability of editing and criticizing the content published in the system. On one hand, the content management environment has to provide users with tools to easily and safely edit the content, and on the other, to help other users develop better quality content pieces (for example, through the use of comments and suggestions, which can be publicly or privately viewed).

Promoting Evolution of Knowledge	**1**	**2**	**3**	**4**	**5**
Collaborative editing of content	Y	Y	Y	Y	N
Mechanism for content criticism	Y	N	N	N	N
Option to sign a piece of criticism or to leave it anonymous	Y	N	N	N	N
Version control	N	N	Y	Y	N
Mechanism for checking in/checking out documents for editing	Y	N	Y	Y	Y
Feedback mechanism to allow users to evaluate the system	Y	Y	N	N	N

Table 8.2: Promoting Evolution of Knowledge

3. Promoting Sharing of Knowledge. This part of the evaluation concentrates on how easy it is for different users to find relevant content on the site and, once they found something that might be interesting to other users, suggest these documents to them. In particular, different knowledge management systems are evaluated on how well they solve the problem of sharing information within a group of users with similar interests and among various groups, members of which might be interested in different types of content.

Promoting Sharing of Knowledge	**1**	**2**	**3**	**4**	**5**
Inter-group sharing and document control mechanism	Y	N	Y	Y	Y
Intra-group sharing and document control mechanism	Y	Y	Y	Y	Y
Ability to assign viewing permissions distinct from modification permissions	Y	N	Y	Y	Y
Ability to suggest potentially interesting relevant content	Y	N	N	N	Y

Table 8.3: Promoting Sharing of Knowledge

4. Promoting Effective Use of Knowledge. The last criterion that the designers of knowledge management systems have to address is how to help users find and personalize the content that is most relevant to their jobs or interests. In general, it deals with different ways to search for information, viewing content in different formats, and the ability to personalize a user's workspace (e.g., through the use of bookmarks or personalized annotations).

Promoting Effective Use of Knowledge	**1**	**2**	**3**	**4**	**5**
Content evaluation	Y	N	N	N	N
Filtering content according to user's personal preferences	Y	N	Y	Y	Y
Bookmarking	Y	N	N	Y	N
Basic keyword search capabilities	Y	N	Y	Y	Y
Advanced search	N	N	Y	N	Y
Conversion of content format	N	N	Y	N	Y
Ability to annotate content	Y	N	N	N	N
Ability to easily connect multiple related objects for private or public use	Y	N	N	Y	Y

Table 8.4: Promoting Effective Use of Knowledge

As evident from the tables presented above, most systems concentrate on one or two goals and either completely ignore the others or do a poor job of addressing them. For example, out of all evaluated systems, only Intranets.com and Know-and-Share promote creation of knowledge by allowing any user within a particular group to contribute to the content present on the site. However, Intranets.com does not allow effective use or evolution of the content. SiteStation, on the other hand, provides effective mechanisms to collaborate and use the content, but does not allow regular users of the site to publish new and innovative ideas. Xpedio and Novation mainly concentrate on giving users an easy and effective way to view and share published content, but restrict users from creating and modifying the content (mostly because they require special software to do so). Overall, only the Know-and-Share system addresses all four issues and provides all users of the system (not only designated content generators and Web site administrators) with an easy way to create, evolve, share and use knowledge.

8.2 Ability to Accommodate Bridge Group Constraints

Know-and-Share is a complex system that allows organizations to create, manage, and share information with other organizations. So how do we measure the results that have been achieved by building this knowledge management system? We need to consider the requirements imposed by the organization that would use our system, the Bridge Group Community, and evaluate if the Know-and-Share environment fits these requirements better than any other existing system available to them.

Many nonprofit organizations such as the Bridge Group don't have time or resources to deal with expensive and difficult to maintain Knowledge Management systems. Our system is targeted towards organizations with limited resources which need to propagate their ideas to a large number of people. The best way for them to achieve the goal of sharing their knowledge with others is to use the power of the Web and communicate their ideas without being in close proximity of each other. The Bridge Group faces several challenges to accomplish their goal of sharing knowledge due to the nature of nonprofit organizations. We have interviewed the Bridge Group management and came up with a set of limitations documented below.

- Limited bandwidth -- most nonprofit organizations do not have state of the art software. Thus, in order to connect to the Internet, they use a modem, which limits their abilities to download files from the Internet. In most systems, modification of documents happens through downloading files from the Internet, editing them, and uploading them back. Online editing would solve this problem for the organizations that face the problem of limited bandwidth.

- Limited time to spend administering the system -- due to limited resources, most nonprofit organizations would not be able to afford a full time administrator of the content posted on the web site by the organization's employees. Thus, the distribution of responsibilities and permissions is very important.

- Need for inter-organizational information sharing -- one of the major goals of nonprofit organizations is to distribute and share information across organizations. A knowledge Management system should provide enough capabilities to allow organizations to share their knowledge with other organizations.

- Lack of technically-savvy personnel to devote to day-to-day administration -- the site should require very little knowledge of the underlying structure. Thus, a non-technical person should be able to handle day-to-day administrative tasks.

- Need to handle a wide variety of Knowledge Types -- the Bridge Group will have many different nonprofit organization. Each organization will want to contribute different types of knowledge. Thus, the system should be able to create a wide variety of Knowledge Types to accommodate different needs of nonprofit organizations.

- Need to ensure protection of sensitive information -- due to the nature of consultant/client relationship of the Bridge Group and other organizations, it is crucial that the system has mechanisms to ensure that the sensitive information contributed to the site is protected. This should be accomplished by an elaborate authorization model.

- Need to accommodate users using different platforms for authoring content -- the Bridge Group clients will use different computer platforms. This presents a problem of downloading documents and editing them offline.

Below is the table that supports our claim of building a system that satisfies all of the above requirements, and other existing systems available for the Bridge Group are not suitable for an organization of its type. While satisfying the above constraints, the Know-and-Share system does not compromise the qualities of an adequate Knowledge Management system as was described earlier in this section.

Addressing Bridge Group's Requirements	**1**	**2**	**3**	**4**	**5**
Limited bandwidth	Y	N	N	N	N
Limited time to spend administering the system	Y	Y	N	Y	N
Need for inter-organization information sharing	Y	N	Y	Y	Y
Lack of technically-savvy personnel to devote to day-to-day administration	Y	Y	Y	Y	Y
Need to handle a wide variety of knowledge types	Y	N	Y	N	Y
Need to ensure protection of sensitive information	Y	N	Y	Y	Y
Need to accommodate users using different platforms for authoring content	Y	N	N	N	N

Table 8.5: Addressing Bridge Group's Requirements

The Know-and-Share system is obviously not suitable for every organization. Some organizations don't care about the cost of the software or time constraints to administer the site. This system is intended for those organizations that do care about the things mentioned above and, based on the research done for this thesis, there are no other existing systems that satisfy all the requirements presented here.

Chapter 9

Conclusion

Knowledge Management is one of the most active areas of research today. Many questions remain unanswered. Our experience building the Bridge Group Community System served as a great introduction to the topic of Knowledge Management. In this final section of the paper, we will outline the future work we feel would benefit the system and conclude by summarizing what we have accomplished and what we gained from the experience.

9.1 Future Work

In the comparative analysis of features presented in an earlier chapter, it was shown that, relative to currently available state-of-the-art Knowledge Management Environments, the Bridge Group Community System offers its users a wide range of features and tools that ultimately serve to satisfy the requirements posed by the Bridge Group. However, there are some mechanisms and features that have not been implemented due to time constraints. Making these enhancements will improve the system by placing more tools at the users' fingertips and making knowledge creation, evolution, sharing and use easier and more intuitive. The following list of desired improvements and addition is organized by the same criteria that were used in the comparative analysis section of this paper.

9.1.1 Promoting Knowledge Creation

Making Object Creation as Easy as Possible

Rather than make the user fill out the entire "Create an Object" form, the system can attempt to assist the user by filling out some of the fields for him/her. For example, if one of the Knowledge Fields of an article object is a file with the actual article, the system can read the file and infer the title, author, and publication date of the article, freeing the user from the need to fill in that information.

The task of partially filling out the form for the user is certain to be challenging since the system is based on user-defined Knowledge Types. However, implementing this "user assistant" mechanism has the potential of significantly improving the user's experience and promoting new knowledge creation.

Approval of Newly Created Objects

To prevent low quality content from being added to the knowledge base, the system should support requiring Objects of a particular Type to be approved before they become available for viewing. This feature was originally included in the system prototype. However, requiring Objects of a certain type to be approved by a particular user (or user group) will prevent users from creating proprietary objects of that type, thus discouraging reuse of Knowledge Types. For example, if a user wants to create an Article that will only be available for viewing to several people, he/she will most likely need to create a new Article Knowledge Type.

Since promoting reuse of Knowledge Types is one of the primary requirements of the Know-and-Share system, the Approval Requirement Mechanism was removed. In the future, however, a better designed Approval Requirement feature should be implemented.

Multiple Inheritance

It would be useful to be able to create an Object that belongs to more than one type, inheriting the Fields that belong to both. This concept, however, involves answering several important design questions. It needs to be decided whether the Object is somehow associated with each Type after creation of whether a separate copy is created for each Type. Or, maybe, a single copy should be maintained that is cut off from all Types except one which is designated as primary?

The current design of the system models the Knowledge Type - Object hierarchy as a tree, which is not a very realistic way of representing the realm of knowledge. Multiple inheritance may offer a solution to this problem.

Creating New Types Similar To Existing Types

Although a lot of effort went into making the process of creating a new Knowledge Type as easy for the user as possible, further improvement is strongly desired. The system should support a creation of a Knowledge Type that "is like" an existing Object Type or several existing Object Types. Such a mechanism should offer the user to pick the Knowledge Fields of the existing Type(s) the new Type should inherit from. New fields can then be added. After the creation process is complete, the newly created Knowledge Type becomes a separate entity.

9.1.2 Promoting Knowledge Evolution

Object Versioning

An important goal of any Knowledge Management system is promoting continuous evolvement of knowledge. Version control helps facilitate understanding of what the Object goes through to reach its current state. When an Object is modified, the old version continues to be stored. When a user views an Object, the latest version is automatically displayed, but the user should have the capability to view an older version if the need arises.

When an Object is deleted, a version should be kept in the system. There should be a way of marking the Object as deleted without physically removing it from the database. The user should be given an option to restore (undelete) the object at any point. Thus extension to the Know-and-Share system, which currently stores only the most recent version of an object, will be very useful.

9.1.3 Promoting Knowledge Sharing

Connecting the Chat and Discussion Board modules to the Know-and-Share system

Presently, the Chat and Discussion Board modules aren't well connected to the Know-and-Share system. Sure, users can use the Discussion Forum to share their experience and tell others what they think about a particular Knowledge Object, but there is currently no way to link a discussion thread to an object. Better integration of the Know-and-Share with other modules of the Bridge Group Community system would make it easier for the users to share knowledge.

9.1.4 Promoting Knowledge Use

Statistics and Reporting

The system doesn't currently provide any tools for generating reports or monitoring usage statistics. In future, a module to allow reporting on user behavior patterns should be included. For example, it would be useful to see which Objects are not being viewed at all or which Groups are most active in development of content.

Another function of the Statistics and Reporting tool would be to provide a user involved in contributing content with a detailed report about the usage the knowledge he/she authored.

Object Expiration Dates

It would be useful for the users to be able to specify an Object Expiration Date. After the expiration date passes, the Object should be marked as Expired. Such a feature would help avoid accumulation of content that is no longer relevant or useful, instead of physically deleting it.

Merging of Knowledge Types

The system should support a mechanism for merging of similar Knowledge Types and Objects that belong to those types. While the system was designed with Object Type reuse in mind, some user-induced redundancy is inevitable. Merging Type B into Type A should probably convert the corresponding fields, insert default values into fields present in Type A but not Type B, and drop the fields that are present in Type B but not in Type A.

Advanced Search Tool

Know-and-Share includes a very simple search tool. As the number of Objects grows, a need for increasingly complex searches will arise. To satisfy this requirement, a more advanced search tool has to be developed.

The Advanced Search tool should provide the following capabilities:

- support for boolean logic (i.e. "find everything with (apple and orange) not pear")
- searching selected Knowledge Types (i.e. "find an Article with (boy or girl)")
- searching by the date when the Object was created or the date when it was last modified
- searching by Object author name
- accompanying the search results by scores that reflect how well the Object matches the search conditions provided by the user
- offering the user to save the result set for future use
- sharing of result sets

It is important to note that implementing the sharing of result sets will involve a lot of thinking about security, authorization, and avoiding storage of a large number of redundant result sets.

9.2 Work Completed

Many commercial Knowledge Management systems exist on the market today. They address a wide variety of needs, but none of them meets Bridge Group's requirements —— none of them provides the look and feel of a community while encouraging all users to author content and supporting collaborative editing and sophisticated authorization schemes. Furthermore, none of the systems can accommodate the constraints that companies in the nonprofit sector are likely to have to deal with —— limited funds to spend on software, limited administrative resources, a wide variety of authoring environments.

As a result of this project, a system that satisfies the requirements listed above has been implemented. The Bridge Group Community environment encourages all users to participate in the knowledge creation process. The system provides extensibility -- each user has the power to create a Knowledge Type with a newly-defined structure if none of the existing types match his/her needs. The Bridge Group Community system provides tools for collaborative development of content -- it includes tools allowing users to criticize, bookmark, rank, and filter existing content. Inter-group as well as intra-group sharing is supported by an extremely flexible authorization model. The system provides scalability through encouragement of reuse of existing Knowledge Types. Administration is distributed among the client user groups, thus making the job of the site administrators easier and less time-consuming, while giving users more control.

The Bridge Group Community system provides an opportunity for the Bridge Group and its clients to collaboratively develop knowledge and share it within as well as among their respective organizations. We have conducted an experiment with the Bridge Group employees, gathered their feedback, made necessary changes based on their suggestions, and discussed future improvements and enhancements. Based on the results of this experiment, we feel that the project was a success despite the fact that, due to time constraints, we did not make the system available to the Bridge Group Clients in time to analyze the results of an experiment with a large number of users in this paper.

We have learned a lot in the course of building the Bridge Group Community. We found out about the unique constraints faced by companies in the nonprofit sector and about the way knowledge is developed and managed in consulting organizations. Most importantly, we learned a lot about challenges one faces when developing a system for a client who does not have a clear understanding of technology. In conclusion, our system has been accepted and initially tested by the Bridge Group, and they intend to present it to its clients shortly.

Bibliography

[1] Bereiter, Carl. Where is the Knowledge in "Knowledge Media"? KMDI Informal Seminar Series, 1997.

[2] Ikeji, Augustine Chidi & Fotouhi, Farshad. An Adaptive Real-Time Web Search Engine. ACM Digital Library.

[3] De Long, Dave and Miller, Heather. Caselets from Knowledge Management Interviews. The Ernst and Young Center for Business Innovation.

[4] Drexler, K. Eric. Hypertext Publishing and the Evolution of Knowledge. Hypertext 87 conference.

[5] Widyantoro, Ioerger, Yen. An Adaptive Algorithm for Learning Changes in User Interests. Texas A&M University, Department of Computer Science. ACM Digital Library.

[6] Chen, Liren and Sycara, Katia. WebMate: A Personal Agent for Browsing and Searching. Carnegie Mellon University. ACM Digital Library.

[7] Wang, Weigang and Rada, Roy. Structured Hypertext with Domain Semantics. ACM Digital Library.

[8] Jones, Steve and Staveley, Mark. Phrasier: A System for Interactive Document Retrieval Using Keyphrases. ACM Digital Library.

[9] Golovchinsky, Price and Schilit. From Reading to Retrieval: Freeform Ink Annotations as Queries. ACM Digital Library.

[10] Cohen, William. A Web-based Information System that Reasons with Structured Collections of Text. AT&T Labs Research.

[11] Sena, James & Shani, A. B. Intelligent Systems: A Sociotechnical Systems Perspective. ACM Digital Library.

[12] Vishik, Claire & Whinston, Andrew. Knowledge Sharing, Quality and Intermediation. ACM Digital Library.

[13] KM World Online <http://www.kmworld.com>

[14] Intranets.com Home Page <http://www.intranets.com>

[15] SiteDriver Home Page <http://www.sitedriver.com>

[16] FatWire Home Page <http://www.fatwire.com>

[17] Novation Product Page <http://www.cimagenovasoft.com/web/products/products.htm>

[18] Xpedio Content Publisher <http://www.intranetsolutions.com/pdf_files/brochure/cpfact.pdf>

[19] Executive Overview of IntraNet Solutions, Xpedio 2.0. Doculabs, 2000. <http://www.intranetsolutions.com/pdf_files/papers/dlabsxcx.pdf>

[20] IBM Knowledge Management. <http://www−4.ibm.com/software/data/knowledge>

Appendix A

Data Model

```
-- The Data Model for the Knowledge Management Center module
-- /www/doc/sql/km.sql
--
-- by Yevgeniya Zemlyakova, Lyudmila Zemlyakova, Sofya Pogreb
--
--
---------------------------KM_OBJECT_TYPE-------------------------------
-- the most fundamental piece of knowledge is the object type

-- sequence to generate object_type_ids
create sequence km_tables_sequence start with 1;

create table km_metadata_object_types (
object_type_id integer primary key,
-- each object type is associated with a table_name which contains
its objects
table_name varchar(21) unique not null,
pretty_name varchar(100) not null,
pretty_plural varchar(100),
approved_by     references users,
-- should objects of this type be approved before users can view them?
needs_approval_p char(1) default ’f’ check (needs_approval_p in (’t’, ’f’)),
modified_by not null references users,
created_by references users
);
```

```
############################SAMPLE STATEMENTS#############################

insert into km_metadata_object_types
(object_type_id, table_name, pretty_name, pretty_plural, modified_by,
needs_approval_p, created_by)
values
(21, 'km_lit_search', 'Thesis Literature Search', 'Thesis Literature Search', null,
'f', 7);

---------------------------------------------------------------------------
------------------------------KM_METADATA_ELEMENT--------------------------

-- each object type is associated with a collection of km_metadata_elements
-- each element will be come a column in the table created for the object type
-- each object of object type A will have all elements associated with
-- type A plus some default elements included in all objects


-- sequence to generate metadata_ids
create sequence km_metadata_id_sequence start with 1;

create table km_metadata_elements (
metadata_id integer primary key,
-- the object type this element belongs to
object_type_id not null references km_metadata_object_types,
-- the name of the column that will contain this element
column_name varchar(30) not null,
pretty_name varchar(100) not null,
-- for example, text or boolean
abstract_data_type varchar(30) not null,
-- this datatype will be used to creat the new column
oracle_data_type varchar(30) not null,
-- sql options -- for example: "not null"
extra_sql varchar(4000),
-- when the user creates/edit objects, how should this element be
-- presented? Example: textarea, checkbox
presentation_type varchar(100) not null,
-- example: maxsize, size, rows=, or an sql statement
presentation_options varchar(4000),
-- should we make the user fill in a value for this element?
mandatory_p char(1) check (mandatory_p in ('t', 'f')),
-- where will this element appear in the form presented to user
form_sort_key integer,
-- should data from this field show up in search results?
include_in_ctx_index_p char(1) check (include_in_ctx_index_p in ('t','f')),
default_value  varchar(200),
-- give the user some pointers as to how he should fill out the field
entry_explanation varchar(4000),
-- if the entry_explanation is not enough....
help_text varchar(4000),
modified_by not null references users,
unique(object_type_id,column_name)
);



-- the KM Center module is centered around the idea of sharing information
-- with a group of people (of course, the group may just have 1 user or
-- it may include all system users)
-- permissions to add/modify/delete elements and modify/delete the type itself
create table km_object_type_permissions (
object_type_id not null references km_metadata_object_types,
-- 'c': objects can be created by this user group
-- 'm': elements/itself can be created/modified by this user group
-- 'd': elements/itself can be deleted by this user group
permission_user_group not null references user_groups(group_id),
permission_type char(1) check (permission_type in ('c','m','d'))
);

##############################SAMPLE STATEMENTS#####################
insert into km_metadata_elements
(object_type_id, metadata_id, column_name, pretty_name, abstract_data_type, oracle_
values
(51, 446, 'column1', 'Select', 'sel', 'varchar(200)', '', 'select', 'multiple size=

----------------------------------------------------------------------
-----------------------------KM_OBJECT---------------------------------

-- while the object type is the fundamental piece of knowledge, the object is
-- what the users will deal with most often

-- sequence to generate object-ids
-- note that we are using the same sequence for objects of all types
create sequence km_objects_sequence start with 1;

-- this will be the only data that connects objects of different types
create table km_all_objects (
km_object_id integer primary key,
object_type_id not null references km_metadata_object_types
);
```

```
-- permissions for a particular object
create table km_objects_permissions (
km_object_id not null references km_all_objects,
-- 'v' objects can be viewed by this user group
-- 'm' elements/itself can be created/modified by this user group
-- 'd' elements/itself can be deleted by this user group
permission_user_group not null references user_groups(group_id),
permission_type char(1) check (permission_type in ('v','m','d','a'))
);

############################SAMPLE STATEMENTS############################

insert into km_all_objects
(km_object_id, object_type_id)
values
(113, 51);

insert into km_objects_permissions
(km_object_id, permission_user_group, permission_type)
values
(112, 254, 'v');

-------------------------------------------------------------------------
-------------------------------LINKS----------------------------------------

-- this table tells us which object_types a particular object can link to
-- links are unidirectional -- thus, saying type A can link to type B is
-- not the same as saying that type B can link to type A
create table km_object_type_map (
object_type_id1 not null references km_metadata_object_types,
object_type_id2 not null references km_metadata_object_types,
primary key(object_type_id1, object_type_id2)
);

-- here are the actual object-object-links
-- again, note that links are unidirectional
create table km_object_object_map (
-- object the user is linking from
object_id_a not null references km_all_objects,
-- object the user is linking to
object_id_b not null references km_all_objects,
-- the user may want to comment on the link and how the two object relate
-- to each other
map_comment  varchar(4000),
created_by not null references users,
creation_date  date default sysdate not null,
-- should the link be visible to other users?
public_p char(1) check (public_p in ('t','f')),
primary key (object_id_a, object_id_b, created_by, public_p)
);

-- index the table to improve performance
create index km_object_object_map_idx1 on km_object_object_map (object_id_a, object
create index km_object_object_map_idx2 on km_object_object_map (object_id_b, object

#############################SAMPLE STATEMENTS#######################
insert into km_object_type_map
(object_type_id1, object_type_id2)
values
(53, 10);

insert into km_object_object_map
(object_id_a, object_id_b, map_comment, created_by, creation_date, public_p)
values
(27, 8, 'company''s website', 7, sysdate, 'f');

--------------------------------------------------------------------
------------------------------COMMENTS------------------------------

-- comments:
-- each user can comment on each element of each object
create table km_metadata_comments (
-- the object the comment applies to
object_id not null references km_all_objects,
-- the field the comment applies to
metadata_id references km_metadata_elements,
column_name varchar(30),
-- the comment author
created_by not null references users,
created_date date default sysdate not null,
-- should this comment be visible to other users
public_p char(1) default 'f' check (public_p in ('t','f')) not null,
-- if other users view the comment, do you want it to be signed with your name?
author_p char(1) default 'f' check (author_p in ('t','f')) not null,
-- the actual comment
user_comment varchar(4000),
-- each user can only comment on a particular piece of knowledge once
primary key (object_id, column_name, created_by)
);
```

```
###########################SAMPLE STATEMENTS############################

insert into km_metadata_comments
(object_id, metadata_id, column_name, created_by, public_p, author_p, user_comment)
values
(101, null, 'keywords', 7, 't', 't', 'wow! interesting keywords');

-------------------------------------------------------------------------
-----------------------------RANKS---------------------------------------

-- ranking is a way for each user to mark some sites as more interesting and others
-- it also lets users benefit from others' experiences -- they can sort objects by

-- sequence to generate rank_id's
create sequence km_object_rank_id_sequence start with 1;
-- this table maps an integer rank_id to a more elaborate rank description
create table km_object_ranks (
km_object_rank_id integer primary key,
km_object_rank_description varchar(30) not null,
-- defines the order ranks should be presented in (this allows us to support addit
km_rank_order integer
);



-- keeps track of each user's ranks
-- note that this table only has rows for those user/object pairs that have a rank
-- this is more efficient in terms of storage space than keeping a row for each pa
create table km_user_object_ranks (
user_id not null references users,
km_object_id not null references km_all_objects,
km_object_rank_id references km_object_ranks,
-- a user can edit a ranking of an object, but can't have more than one ranking fo
primary key (user_id, km_object_id)
);

create index km_user_object_ranks_idx1 on km_user_object_ranks (user_id);
create index km_user_object_ranks_idx2 on km_user_object_ranks (km_object_id);

#########################SAMPLE STATEMENTS############################
insert into km_object_ranks
(km_object_rank_id, km_object_rank_description, km_rank_order)
values
(6, 'Liked it', 3);
```

```
insert into km_object_ranks
(km_object_rank_id, km_object_rank_description, km_rank_order)
values
(7, 'It''s ok', 2);

insert into km_object_ranks
(km_object_rank_id, km_object_rank_description, km_rank_order)
values
(8, 'Disliked it', 1);

------------------------------------------------------------------------
--------------------------------BOOKMARKS-------------------------------


-- each person is likely to use a certain subset of objects more often than the res
-- bookmarks provide a utility for easily accessing the objects one uses most
-- each user has a personal set of bookmarks
create table km_user_object_bookmarks (
user_id not null references users,
km_object_id not null references km_all_objects,
primary key (user_id, km_object_id)
);

##############################SAMPLE STATEMENTS########################

insert into km_user_object_bookmarks
(km_object_id, user_id)
values
(101, 7);

-------------------------------------------------------------------------
--------------------------------OBJECT_TYPE EXCLUSION-------------------

-- certain object_types may not be interesting to all users
-- (even the users who have permissions to view the object type)
-- object_type_exclusion lets users mark an object type as "non-preferred"
-- the non-preferred object types will be listed below the preferred
-- object types in object listings
create table km_user_object_type_exclusion (
-- object_type exclusion is personalized
user_id not null references users,
object_type_id not null references km_metadata_object_types,
-- each user may only exclude each object type once
primary key (user_id, object_type_id)
);

##############################SAMPLE STATEMENTS########################

insert into km_user_object_type_exclusion
(user_id, object_type_id)
values
(7, 6);


------------------------------------------------------------------------
--------------------------------SUBJECTS--------------------------------

-- subjects for the site to classify objects

-- sequence to generate km_subject_id
create sequence km_subject_id_sequence start with 1;

-- contains all subjects
create table km_subjects (
km_subject_id integer primary key,
description varchar(100)
);

##############################SAMPLE STATEMENTS########################

insert into km_subjects
(km_subject_id, description)
values
(1, 'Best Practicies');


------------------------------------------------------------------------
-------------------------------OBJECT_SUBJECT_MAP--------------------------

-- each object can belong to one or more subjects
-- table that maps an object to a subject; many-to-many relationships
create table object_subject_map (
subject_id not null references km_subjects,
object_id not null references km_all_objects
);

##############################SAMPLE STATEMENTS########################

insert into object_subject_map
```

```
(subject_id, object_id)
values
(1, 27);

--------------------------------------------------------------------------
------------------------------SUBJECT EXCLUSION---------------------------

-- certain subjects may not be interesting to all users
-- subject_exclusion lets users mark a subject as "non-preferred"
create table km_subject_exclusion (
-- subject exclusion is personalized
user_id not null references users,
subject_id not null references km_subjects,
-- each user may only exclude each subject once
primary key (user_id, subject_id)
);

##############################SAMPLE STATEMENTS########################

insert into km_subject_exclusion
(user_id, subject_id)
values
(5, 1);

-------------------------------------------------------------------------
---------------------------------USER_PROFILES---------------------------

-- contains user profiles (optional to users)
create table user_profile (
user_id not null references users,
update_date date default sysdate not null,
organization varchar(200),
city varchar(200),
comments varchar(4000),
hint varchar(200),
-- does the user want to be emailes about changes to the site
notification_p char(1) check (notification_p in ('t','f'))
);

##############################SAMPLE STATEMENTS########################

insert into user_profile
(user_id, update_date, organization, city, comments, hint, notification_p)
values
(5, 2000/05/01, 'MIT', 'Cambridge', 'I helped to develop this site',
'my friends', 't');

---------------------------------------------------------------------------
---------------------------------------------------------------------------


-- The Data Model for the Survey module
-- /www/doc/sql/survey.sql
--
-- by Yevgeniya Zemlyakova, Lyudmila Zemlyakova, Sofya Pogreb
--
--

-- sequence to generate survey id
create sequence survey_id_sequence;

-- table that contains basic information about a particular survey
create table surveys (
survey_id integer not null primary key,
name varchar(100) not null,
description varchar(4000),
-- make the dates NULL for an on-going survey
start_date date,
end_date date,
-- should the user be a registered user to take this survey
require_registration_p char(1) default 'f' check
(require_registration_p in ('t','f')) not null
);

--#############################SAMPLE STATEMENT############################
insert into surveys
(survey_id, name, description, start_date, end_date, require_registration)
values
(41, 'The Bridge Group Site Feedback Survey', 'Please fill out the
following survey to let us know what you thought about this
site. Any suggestions you can make are very welcome.',
sysdate, '2000-05-26', 't')

--########################################################################


-- sequence to generate question id
create sequence question_id_sequence;

-- table that contains basic information about a particular question
```

```
create table questions (
question_id integer not null primary key,
name varchar(100) not null,
description varchar(4000),
-- order that the question will be presented
sort_order integer,
-- type of a question: radio buttons, checkboxes, open-ended
questions
type char(1) default 'r' check (type in
('r','b','o')) not null,
-- if the question is of type radio or checkboxes, will it have
-- field 'other, please specify' at the end
other  char(1) default 'f' check (other in ('t','f'))
not null,
-- should the results of this question be public for other
people to see
private  char(1) default 'f' check (private in ('t','f'))
not null,
-- maximum length of 'other' field
maxlength integer default 200 not null
);

--###########################SAMPLE STATEMENT###########################

insert into questions
(question_id, name, description, sort_order, type, other, private,
maxlength)
values
(84, 'User Background', 'Please tell us a little about yourself', 1,
'b', 't', 'f', 500)

--######################################################################


-- maps a question to a survey
create table survey_question_map (
survey_id references surveys not null,
question_id  references questions not null
);

-- index the table to speed up performance
create index question_survey_index on survey_question_map(question_id);
create index survey_question_index on survey_question_map(survey_id);

--###########################SAMPLE STATEMENT############################
```

```
insert into survey_question_map
(survey_id, question_id)
values
(41, 84)

--###########################################################################


-- sequence to generate choices for questions
create sequence choice_id_sequence;

-- table that constains information about a particular choice for a
question
create table question_choices (
choice_id  integer not null primary key,
question_id references questions not null,
-- choice label (description for user)
label varchar(500) not null,
-- order in which this choice is presented
sort_order integer,
-- 't' if this choice is none
-- (for users who do not answer this questions)
none char(1) default 'f' check (none in ('t','f'))
not null
);

-- index the table to speed up performance
create index question_choices_index on question_choices(question_id,
choice_id);

--###########################SAMPLE STATEMENT###########################

insert into question_choices
(choice_id, question_id, label, sort_order, none)
values
(82, 84, 'I work for a non-profit organization', 1, 'f')

insert into question_choices
(choice_id, question_id, label, sort_order, none)
values
(111, 101, 'none', null, 't')

--#######################################################################
```

```
-- table that contains choice made by each user for every question
create table question_user_choices (
question_id references questions not null,
choice_id references question_choices not null,
-- user_id can be NULL if we're not requiring registration
user_id references users,
ip_address varchar(50) not null,
choice_date date not null
);

-- index the table to speed up performance
create index question_user_choice_index on
question_user_choices(question_id);
create index question_user_choices_index on
question_user_choices(choice_id);

--###########################SAMPLE STATEMENT##########################

insert into question_user_choices
(question_id, choice_id, user-id, ip_address, choice_date)
values
(93, 90, 5, '18.186.0.24', '2000-04-27')

--######################################################################


-- table that contains information extracted from 'other' field
-- from radio/checkbox types of questions
create table question_user_other_choices (
question_id references questions not null,
-- user_id can be NULL if we're not requiring registration
user_id references users,
ip_address varchar(50) not null,
choice_date date not null,
-- what user wrote in 'other, please specify' field
description varchar(500)
);

-- index the table to speed up performance
create index question_other_choice_index on question_user_other_choices(question_id

--###########################SAMPLE STATEMENT###########################

insert into question_user_other_choices
```

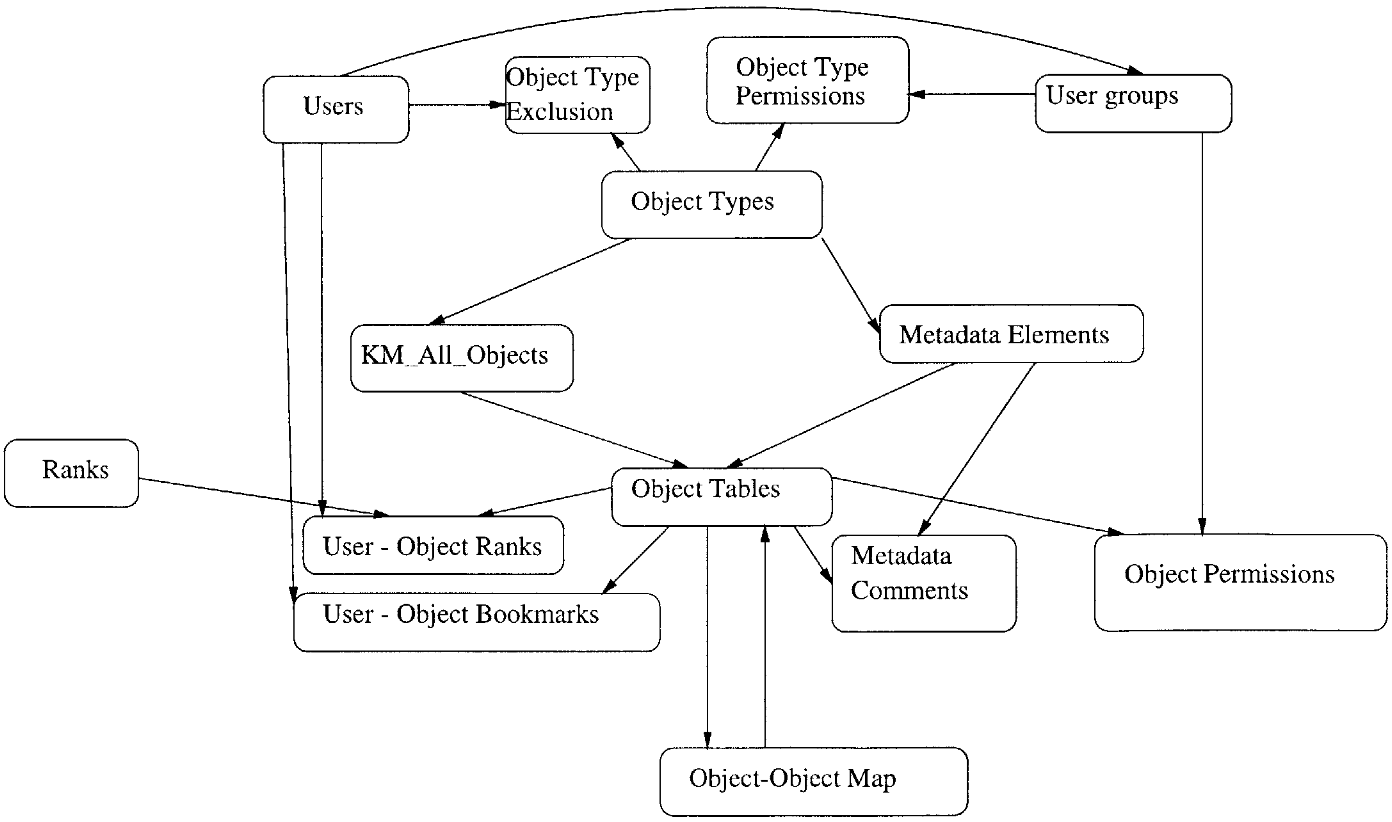
```
(question_id, user_id, ip_address, choice_date, description)
values
(114, 5, '18.186.0.24', '2000-04-27', 'I am a student')

--######################################################################


-- table that contains information extracted from open-ended questions
create table question_user_open_choices (
question_id references questions not null,
-- user_id can be NULL if we're not requiring registration
user_id references users,
ip_address varchar(50) not null,
choice_date date not null,
answer varchar(4000)
);

-- index the table to speed up performance
create index question_open_choice_index on
question_user_open_choices(question_id);

--###########################SAMPLE STATEMENT#############################

insert into question_user_open_choices
(question_id, user_id, ip_address, choice_date, answer)
values
(97, 5, '18.186.0.24', '2000-04-27', 'need more documentation')

--#######################################################################


-- function that checks whether the survey is active
create or replace function survey_is_active_p (start_date in date,
     end_date in date)
return char
as
    result char;
begin
    result := 't';

    if (trunc(start_date) > trunc(sysdate)) then
        result := 'f';
    end if;

    if (trunc(end_date) < trunc(sysdate)) then
```

```
        result := 'f';
    end if;

    return result;

end survey_is_active_p;

/
show errors;
```

Appendix B

System Architecture Diagram

Know-and-Share Architecture



A → B means that data from table A is used in table B