EXTENDING USER MODELS AND DATA FILTERING TECHNIQUES IN ADAPTIVE WEB SITE DEVELOPMENT

THESIS

Presented to the Graduate Council of
Southwest Texas State University
in Partial Fulfillment of
the Requirements

For the Degree

Master of SCIENCE

By

Michael S. Flaherty, B.A.

San Marcos, Texas
December 2003

ACKNOWLEDGEMENTS

I would like to thank my mother, Dr. Julia Flaherty, and my father, Dr. Dan Flaherty, for teaching me the value of education. Their support and understanding that life is a continual learning process made much of my education and this project possible.

I am in deep appreciation to my thesis committee members for their willingness to participate in this process and for their valuable insight. I owe a debt of gratitude to Dr. Ron Sawey and Dr. Carol Hazlewood for their patience, flexibility and encouragement during the final push on this project.

Finally, I owe the most gratitude for the realization of this project to my main advisor, Dr. Greg Hall, whose concern for his students, zest for teaching, and desire to make an impact on students' lives shines through in all he does. His actions taught me that while sometimes our accomplishments may go unnoticed or be undervalued, the true accomplishment is that which we do selflessly and for which we seek no reward. I am in debt to him as much for his wisdom and guidance on this project as for his example.

This manuscript was submitted on August 26, 2003.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The success or failure of a Web site is inextricably tied to the user's ability to find the information he or she needs. Static Web sites, those that do not change based on user interaction, cannot hope to provide the level of engagement possible with more dynamic Web sites. In an age when vast volumes of data are available on-line and a continually expanding number of users are looking to the World Wide Web for commerce, for entertainment and as an information resource, users demand and deserve to be able to more efficiently gather the data they need. Web sites that ignore the problems associated with finding information in a sea of data can hardly hope to survive.

This underscores a substantial need for Web ventures of all types to proactively evaluate the accessibility of data at their sites and to implement processes to improve accessibility. Particularly for large sites with vast amounts of data, the need to scale-down and refine the amount of data to meet the needs of specific users is of utmost importance, if the site wishes to retain users. Adaptive Web sites provide perhaps the most promising approach to meeting the needs of both the user and Web site administrator. It is the generation of these adaptive sites that constitutes the research and development undertaken as part of this research project.

1.1 Information Imperatives

The ever-increasing volume of information on the Web reinforces the need for facilities that aid the user in finding the information they seek and for Web sites to connect users with their data. For the Web merchant, the implications are severe, as users become habituated to the sites they can navigate easily and forego the ones where information retrieval is reduced to following a series of static hyperlinks. For the Web master, organizing Web pages to readily provide their data to all users is a daunting if not unachievable task.

Besides the implications this has for Web commerce, the implications for the user are significant, as the expanding body of information on the Web becomes more and more difficult to navigate in an efficient way. It is not humanly possible for users to sort through huge data stores to locate information. Clearly, some automation in the refinement of data is necessary.

Adaptive Web pages are a primary approach for dealing with these problems. Assisted by the use of data filtering techniques, they typically change the data presented on their pages based on the individual user, or use patterns. In this way, users are presented with a refined set of data tailored to their particular needs. When developing these types of pages, the challenge becomes first to identify meaningful user data and use patterns, and then to exploit that information in the generation of customized pages.

1.2 Research Focus and Contributions

This project considers issues key to adaptive Web site development and cultivates a new understanding of the dynamics involved in the development of adaptive Web pages. Exploring various techniques in use today to assist the generation of these dynamic Web

pages, the research focuses on improving user experience and perceived benefit of these sites while facilitating the development of these adaptive systems. The contributions made through this research are as follows:

- The effectiveness of existing techniques are examined and discussed, and the benefits and drawbacks of these approaches are considered within a context of the overall goal of adaptive system generation.
- A new combinatorial approach is developed to assist the generation of an adaptive system utilizing the existing techniques of content-based filtering, collaborative filtering, and an extended, explicitly-defined, user model.
- The Movie Friend Web site, a prototype adaptive system based on the combinatorial approach is fully implemented to facilitate the testing and validation of the developed combinatorial approach. It utilizes a proper subset of the various techniques used to generate dynamic content and provides an effective solution to the problem.
- A qualitative evaluation of the developed prototype furthers understanding of the dilemmas faced when producing dynamic pages and provides insight for consumers and developers of Web content into the viability of the solution.

1.3 Document Overview

This document first examines the types of adaptive Web sites – whether the site is customized for an individual user or transformed based on all users of the site. It further looks at some current practices of initiating adaptations based on the content of pages or based on users' navigational choices. It then examines research in this area, finding a number of different approaches in use today (with differing strengths and weaknesses)

that seek to facilitate the production of these refined Web pages based on individual users and use patterns.

Focusing on page clustering and the techniques of collaborative and content-based filtering, the thesis examines their efficacy when used individually and combined with other techniques. An analysis of projects currently employing some of the clustering algorithms and filtering techniques lends insight into how these methods enhance navigability and increase user satisfaction with these sites.

Additional research is conducted as part of the thesis that includes the exploration of previous projects that have focused on the combination of the various existing techniques used to implement the adaptive functionality.

The adaptive strategy based on the production of the hybrid combinational approach is described, followed by a discussion of the motivations, procedures, limitations, and design decisions made during the development of the Movie Friend prototype. The procedures, goals and results of the qualitative analysis intended to assess the Movie Friend prototype are then produced and provide valuable understanding of the new adaptive strategy developed and implemented as part of the research effort.

With the overwhelming amount of data on the Web as a whole (and vast amounts just in individual Web sites) users require more efficient ways to access data and commercial enterprises demand better targeting of their data. While other methods exist for improving data retrieval on the Web, adaptive Web pages provide an effective solution that provides businesses with the best return on investment in their Web sites, maximizes user interest, and provides users with the data they are seeking.

CHAPTER 2

PROBLEM DOMAIN AND CURRENT APPROACHES

2.1 Problem Background

From the perspective of the information seeker, the challenges set forth by the ever-increasing vastness of the World Wide Web and the body of information that it conveys is astonishing. The exponential growth in number of Web sites and html pages, the rapid expansion of commercial offerings and business services on the Web, and a global onslaught of users joining the Web community all emphasize the need to aid users in information retrieval.

Web researchers are pointing to the increasing difficulty of navigating the Web to find useful information as it expands. Not surprisingly, the second most commonly cited problem by Web users is the inability to "find the information they seek in a simple and timely manner"(Kobayashi & Takeda, 2000). Others have taken various approaches to address this problem of connecting users with the data they seek. Much of the research has focused on intelligent agents, search engines and machine learning algorithms, with varied levels of success.

Levy and Weld (2000), in their article "Intelligent Internet Systems", divide Internet applications into four categories: "user modeling, discovery and analysis of remote information sources, information integration, and Web site management." These

divisions provide useful perspectives from which to assess Internet systems, but most systems cannot be relegated to just one category, and indeed, the most successful Internet systems employ techniques across these categorical distinctions.

Recent approaches incorporate the lessons learned in previous research, employing user models and extending "recommender" systems via filtering processes to dynamically generate Web pages. These adaptive Web sites are one of the more promising approaches to addressing this problem of information overload (Good, Schafer & Konstan, 1999; Shardanand & Maes, 1995).

2.2 The Case for Adaptive Web sites

The phenomenal growth of the World Wide Web provides significant challenges as a potential marketplace and business resource. According to recent studies, by 2000 there were 2.1 billion "unique, publicly-accessible" pages on the Web with around 7 million being added daily (Pastore, 2000).

While presenting statistics for users on-line is more challenging, as of September 2002, market research estimated that more than 605.6 million people worldwide were on-line (NUA Internet Surveys). In turn, this vast and expanding marketplace is primed to fuel a projected $6.8 trillion in on-line "business-to-business and business-to-consumer transactions" in 2004 (Global Reach).

Clearly, the ultimate success or failure of a Web site - as an informational tool or a global marketplace - is inextricably tied to the user's ability to access its data. When viewing the Internet as a huge knowledgebase (e.g., Levy, et al., 2000) we see that Web management systems are essential to achieving this goal of accessibility. And as more and more sites employ techniques that hold user attention and provide the most desired

data to an individual user, those sites that provide the best user experience are likely to be the ones that survive.

2.2.1 Adaptively-structured sites

Two of the more prolific authors in the study of adaptive Web sites, Mike Perkowitz and Oren Etzioni of the University of Washington, say that the feasibility of adaptive Web sites is unproven. While approaches to generating adaptive sites are varied, Perkowitz and Etzioni argue their structure is typically conceived through the differing approaches of *customization* and *transformation.*

A *customized* Web site structures and displays itself "to the needs of each individual visitor, based on information about those individuals" (Perkowitz, et al., 2000). Whereas, a *transformed* Web site improves "the site's structure based on interactions with all visitors." Examples abound on the Web of sites that use this idea of *transformation*, at least to some limited extent. One need only look at the list of top selling products that adorn many Internet retail Web sites, to see this idea in use. Due to its much more limited use, *customization* is more difficult to spot, but can be found on Web sites such as Amazon.com. Its personal recommendations and display of items (based on user profile) make it one of the more widely recognized and successful Internet businesses employing *customization.*

An array of techniques can be applied to assist the development of these customized and transformational sites. One such technique involves the use of *collaborative filtering.* The *collaborative filtering* method relies on a knowledgebase of user ratings and preferences and seeks to connect users with similar tastes and interests to items/data these similar users prefer. This approach has been employed widely in recent research (e.g.

Balbanovic & Shoham, 1997; Good, et al., 1999; Kohrs & Merialdo, 1999; Melville, Mooney & Nagarajan, 2001; Polcicova & Navrat, 2000).

Another technique relies on the construction of a user model developed based on a user's previous access of a site. This model is derived based on the user's access patterns, and might include static preferences set by the user, if the site allows such, and data collected about the user (e.g., Fernandez, Florescu, Kang, Levy & Suciu, 1998; Fink, Kobsa & Nill, 1998). Users can then be classified in user groups, often referred to as "cliques" or "classes", with a specific set of defined preferences. These defined preferences can then be used to determine how to customize the site's structure for the individual user or transform the site's structure.

Yet another technique, particularly in use in sites taking the transformational approach, is the practice of data mining of Web server logs. The goal is to extract useful data about user access patterns (pages visited, links selected, information requested) and use that data to dynamically change the site's structure (e.g., Mobasher, Cooley & Srivastava, 2000; Perkowitz & Etzioni 1997).

Figure 2.1 below shows a high-order adaptive Web site management system that includes all the techniques for generating adaptive Web sites previously mentioned and some additional ones discussed later in this work.

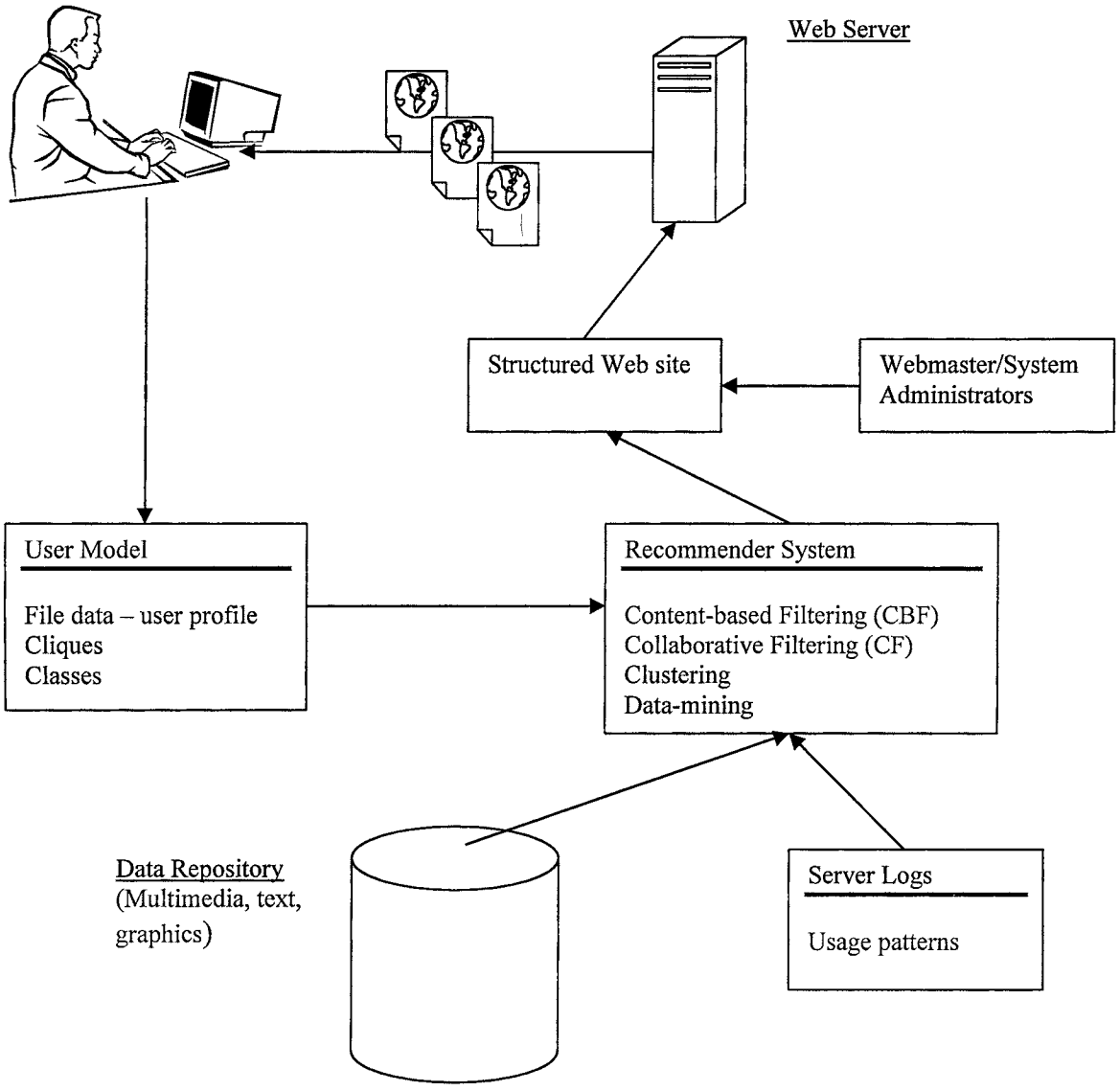


Figure 2-1: A high-order adaptive Web site management system.

For practical purposes, the recommender system in Figure 2.1 encompasses content-based filtering (CBF), collaborative filtering (CF), clustering and data mining. Clustering and data mining are included for simplicity but they could easily be modeled as a separate entity. Still, data mining with the intent of reporting user patterns falls well within the scope of a recommender system and clustering typically employs CBF to some degree.

Nevertheless, it is clear to see that the key component to the adaptive system diagram is the recommender rystem. Without it, the system is relegated to a static existence. The user model and server logs provide useful contributions to an adaptive Web site, but are meaningless without the recommender system. Realizing a distinction exists between the key components of the adaptive Web site, we see that a functional duality is present. Perkowitz and Etzioni (1997) describe this duality as such:

> An adaptive site has two basic components: an *observation module* and a *transformation module*. The observation module monitors user interactions with the site and accumulates important statistics about pages accessed, links traversed, paths followed, and problems encountered. The transformation module draws on this data to make changes to the structure of the site. (p. 4)

Utilizing this distinction we see that the recommender rystem in Figure 2.1 is the transformation module of the system, and the user model and server logs depicted comprise the observation module. This system view is helpful in understanding the role of components within an adaptive system and in determining which adaptive techniques are best suited to that system. It's important to note that few, if any, adaptive sites employ

all the recommender systems techniques shown in the diagram, and many opt to not use server logs or user models in customizing or transforming Web sites.

2.2.2 Statically-structured sites

Unlike adaptively structured sites, statically structured sites make no attempt to customize pages, transform the site's structure or even collect data about users and access patterns. We can approximate a statically structured site utilizing the high-order adaptive Web site management system in Figure 2.1. By removing the key adaptive components - recommender system, server logs, and user model – from the diagram and simply linking the data repository to the structured Web site node we can envision a statically structured site.

Lacking the necessary functionality to make meaningful adaptations to the site, the static site's structure is a product solely of the site developers' initial organization, design and development. No matter how many times a user visits the site, and regardless of who the user is, the static site's pages will always remain the same, providing the same group of links and data on each page.

The design process for such a site inevitably involves a number of assumptions on the part of designers. Assumptions about typical users, access points and patterns yield a rigid, inflexible structure. As the site's complexity increases, the difficulty in producing a site that makes its information readily available expands considerably. Perkowitz and Etzioni (2000) describe the designer's dilemma:

> The designer must anticipate the users' needs and structure the site accordingly. Yet users may have vastly differing views of the site's information, their needs may change over time, and their usage patterns

> may violate the designer's initial expectations. As a result, Web sites are all too often fossils cast in HTML, while user navigation is idiosyncratic and evolving. (p. 152)

Unlike adaptively structured sites, statically structured sites provide few navigational benefits, other than maintaining strict paths through the Web site. This can be helpful for users who perform repetitive tasks at the site or want to retrieve previously located data. In fact, due to the prevalence of statically structured sites on the Web, many users have become habituated to this type of navigation. For this reason, we see that dealing with user expectations is one of the significant challenges to implementing an adaptive Web site design. Beyond this aspect, though, statically structured sites provide little advantage when searching for new data.

To see the true power that an adaptive Web site can have over a static one, consider the computer maker Dell and the "Premier Support" service offered through their Web site, dell.com. This service is intended for administrators of networks, or in general, multiple-computer purchasers of Dell machines, and is a valuable tool in maintaining those machines. Providing system information for all systems purchased, system specifications, device drivers for various operating systems, user manuals and trouble-shooting tools, it provides quite a convenient way to get service and support for Dell computers. Still, since the site is predominantly static, allowing only limited adaptability, the regular user can quickly see where automatic adaptation could make the site a much more powerful tool.

It's feasible to assume that a network administrator might want to download a device driver using a Web browser, for example, install that driver on the local machine, and

then move to a neighboring machine or group of machines to find and install the same driver. This real world example shows where the static approach fails. Requiring an e-mail address and password, the Premier Support option already has enough information to identify the user and make changes to the site's structure based on the user profile and frequent or recent access patterns. Surprisingly, very little of this advantage is exploited.

To perform the example task described above, one must log-in using the Premier Support log-in on each successive machine, search for the computer by serial number or enter the machine model to obtain system information, select a support tool (in this case downloads), select the download category (audio, video, etc...), select an operating system, select a language, select the appropriate link to the download, select the appropriate links through two more pages of download information, select a site from which to download the file and agree to the compliance disclaimer. In all, we've logged-in and made 11 mouse-clicks selecting list items or links to get to the example download, with no consideration for the time required to read pages and determine the appropriate links to select.

Obviously, an adaptation as simple as putting a "top ten" list of links to previously accessed items on a user's home page after log-in would be a huge improvement for users who perform such repetitive tasks. After the first download on the initial machine, a link to the download we just performed would be at the top of our list; so performing the same task on successive machines would mean no more than one or two clicks. The advantage gained by the adaptive site over the static site is clear for this simple example of a repetitive task, but the true power of an adaptive Web site is realized in finding and assimilating new data.

2.2.3 Advantages of adaptive Web sites

At the most basic level, the validity of adaptive Web sites is apparent. Consider a Web site devoted to selling movies and movie memorabilia. With a static site, the Web master is tasked with developing views for the data at the site. For example, if the Web master has the foresight to first divide the site's merchandise by category, he or she might come up with the view *posters* which represents all the movie posters that are available from the Web site. Clearly, this view is inadequate, in and of itself, when the user is seeking a specific poster and when data accessibility is a goal. The *posters* view could result in an index listing of thousands of links to posters. While presumably organized – alphabetically by title, for instance – it still potentially requires the user to navigate through hundreds of pages to get to the title they seek.

Obviously this is an extreme example, but the importance lies in seeing that a statically structured site's views are explicitly defined by the site's structure. For this reason, maintaining a static site is a considerable task, particularly when the site is large, with many pages and varied data. When a new view of the site's data is needed, a static site's Web master must generate it, in terms of creating and adding the appropriate pages and links, etc... This inflexibility makes static sites more costly to maintain and less likely to assimilate new views.

In an adaptive Web site, users are not restricted to existing views of the site, but rather the site retains the flexibility to automatically re-structure based on new views. Of course, these views are based on user information, user actions and filtering techniques. The result is that the site is dynamically structured and much of the maintenance workload required by static sites is eliminated.

Possibly the greatest advantages that adaptive Web sites have over static ones is their ability to maintain user attention and provide a better experience for users. This is notably significant for sites involved in Web commerce. Recent studies have shown that 50% of all Web traffic takes place at the top 900 most popular sites (cited in Kobayashi, et al., 2000). The reasons for why the majority of users visit a certain group of sites is another study entirely, but it's fair to say that user experience and satisfaction play a role, and that being in the top 900 sites would be no detriment to commercial success.

Furthering the example from section 2.2.2, a major part of the network administrator/system buyer's decision to buy a particular brand of computer would presumably be based upon the support. And were the decision between two similar manufacturers down to just the Web site support, it would be hard to deny that an adaptive site might just be the winning point for a big contract. The "top ten" list in the previous section alone would be enough of a time saver for some users to sway them over, and that functionality only scratches the surface of what is possible.

The presumption is that once users are given the opportunity to use these adaptive sites alongside their static brethren, they would come to appreciate the ease-of-use, features and intuitiveness of the adaptive site, and would frequent those sites that provide adaptive functionality.

2.2.4 Disadvantages of adaptivity

Automatic customization does come with significant tradeoffs. Initial development of an adaptive Web site is likely more costly than developing a static one, utilizing more development tools and requiring greater integration of scripting languages for the Web into the site's design and HTML generation. It likely also involves the modification of

the site's data in terms of applying content-based information about each of the files that make-up the site's data. This is required to support content-based filtering (CBF).

To see why this is necessary, consider first a text file. A simple method to extract content-based information from a text file is just to search the contents of the file seeking a pattern match or keyword match. This is not as easy for multimedia data such as images and music. Beyond the usual identifying characteristics (title, artist, song title, etc.), defining the content of a multimedia file may be difficult and not easily obtainable from the file itself. Some researchers (e.g., Kohrs, et al., 1999) have employed "automatic multimedia indexing technologies" to address this. They use these technologies to extract information (color, texture, etc.) about classic paintings in their Active Web-Museum project, described in greater detail in section 2.3.3. Suffice it to say that there may be significant work for an adaptive site in implementing its adaptive functionality and in making the site's data meaningful to that adaptive functionality.

Another key approach, collaborative filtering (CF), suffers from the problem of *sparsity* and the *first-rater* problem. Good (Good, et al., 1999) argues that "for a CF system to work well, several users must evaluate each item; even then, new items cannot be recommended until some users have taken the time to evaluate them". In this statement lies the essence of the *sparsity* and *first-rater* problems. Essentially, the *sparsity* problem is a lack of user ratings for multiple items. Because of this, "the probability of finding a set of users with significantly similar ratings is usually low" (Melville, et al., 2001). And, in terms of the *first-rater* problem, obviously an item must be rated to be recommended. Therefore, it is necessary to employ methods to overcome these key problems when deploying collaborative filtering in a recommender system.

Also, since the adaptive Web site's views are defined by the user and automatically generated by the system, we risk leaving some users confused or frustrated by the adaptive presentation. This problem of user expectations, as mentioned in section 2.2.2, is apparent in users confused by the re-arrangement of pages and links on pages they have visited before. While adaptive functionalities typically produce the same results from the same set of user actions, this is not guaranteed. For this reason, this user expectation - to click down familiar paths at a site - is not easily addressed and is one of the more dynamic considerations in developing adaptive Web sites.

2.3 Current Research

Most researchers in the field draw a distinction between *adaptable* and *adaptive* sites (e.g., Fink, et al., 1998). An *adaptable* site is one that can be manually changed. That is, a site where a user can set static preferences on how the page is displayed - such as setting a stock ticker to appear on the page. In contrast, an *adaptive* Web site is one that collects data from the user, either directly or transparently to the user and then modifies the structure of the Web site such that the information displayed is personalized for that user. In research, these sites are also referred to as automatically adaptive or user-adaptive.

2.3.1 Manually adaptable Web sites

A manually adaptable site allows static user customization. This type of adaptability is common in Portal sites such as Yahoo!, Excite and Lycos. For example, the MyYahoo! services from Yahoo! allow the user to add tools like calendars and stock quotes, to pick individual preferences - such as displaying local weather and daily horoscope - and to organize these custom items on their MyYahoo! page. Both Excite and Lycos offer similar adaptability.

While these sites are adaptable, they are the result of a static, one-time setting of preferences, and as such, are still static sites. Having no dynamic components to automatically re-structure the site, these manually adaptable sites maintain the benefits of static sites and don't fall prey to the disadvantages of adaptivity. They deliver some of the benefits of an adaptive site, providing customized conveniences for users, but they don't seek to exploit the power promised by a truly adaptive site.

2.3.2 Automated adaptive Web sites

Research in building adaptive Web sites has been evolving since the mid-1990s. Early approaches focused mostly on collecting users' use patterns and saw these patterns as the driving force in developing adaptive sites. Even so, the result of this pattern collecting equated initially to making structure "suggestions" to the Web master, who would then choose to implement or not implement the implied changes to a static site's structure.

Later approaches sought to use these user access patterns to truly automatically adapt the site (see Perkowitz, et al., 1997). What is apparent is that this pattern discovery process initiated the classification of users as "types" - i.e., certain "types" of users do certain things. This led to the development of user models, recommender systems and the realization of content-based filtering (CBF) and collaborative filtering (CF). The latest research has sought to combine many of these techniques to reap the benefits of each, with varied success.

2.3.2.1 Collecting use patterns

As mentioned in the previous section, collecting use patterns was the initial approach taken by many researchers in the development of adaptive Web sites, and still remains a

key component to many adaptive systems. This process of collecting user patterns takes the form of parsing Web server usage logs, with the aid of software services such as WebThreads, which allows the Web server to identify individual users and record their paths through a site (see Perkowitz, et al., 1997). Data derived from server logs can then be used with various data mining techniques to generate customized pages.

Mobasher, Cooley and Srivastava take this approach in developing the ACR News site (2000). They define *user transactions* as units of user activity meaningful to data mining and describe three key Web usage-mining techniques: *transaction clustering*, *usage clustering* and *association rule discovery*. These three techniques allow the system to group related user transactions, group related usage patterns, and realize associations between items, respectively. The difference between *transaction clustering* and *usage clustering* is slight, but notable, in that it allows for the generation of some differing views of data. For example, clustering by usage patterns ignores transactions, and thus, allows a view across multiple transactions. Conversely, clustering by transaction disregards usage, and thus, allows the view across usage patterns.

A similar approach (see Perkowitz, et al., 2000) focuses on cluster mining as the second module in the IndexFinder system. This system is designed to automatically generate index pages - which are pages of related links. The first module in the system focuses on processing server logs to produce meaningful data, and after data mining, continues with the third module of conceptual clustering. While the ACR News site undertakes some additional and complimentary data mining techniques as part of the cluster mining stage, the simplified architecture of the IndexFinder system is a good

example of these automated adaptive systems based on usage patterns, and is representative of common efforts that take this approach.

Initial efforts also carried forth the concepts of *promotion* and *demotion* in relation to links or pages (Perkowitz, et al., 1997). In effect, items are promoted or demoted based on a constantly updated popularity score. Simply stated, the more popular items in a view would gravitate towards the top of any index listing, page, etc.

Mining server logs has failings, though, in terms of sacrificing anonymity. Many users are sensitive about having data collected about them, and user privacy issues are a continuous concern on the Web. Data mining need not be inherently intrusive, as demonstrated by the Footprints concept (Wexelblat & Maes, 1997). Under this concept, the activity of users is monitored by the system, but the system doesn't attempt to identify nor match profiles to individual users. In this system, common paths through the Web site, in terms of common traversals of links at the site, dictate the site's structure. These common selections are identified and users are directed to well-worn paths when visiting the site. This is based on the belief that users typically desire the same types of data.

The technique utilization summary for research projects table in Figure 2.2 shows the various techniques used in each of the research projects in adaptive Web site development discussed in this work. It provides a glimpse of the combinations of techniques used to achieve adaptivity.

Research Project	Log processing	Cluster mining	Conceptual clustering	User model	Content-based filtering (CBF)	Collaborative filtering (CF)
ACR News	✓	✓	✓			
Active WebMuseum					✓	✓
AVANTI				✓		
COBWEB		✓	✓		✓	
FAB					✓	✓
IndexFinder	✓	✓	✓		✓	
Movie Friend*				✓	✓	✓
Ringo						✓

* Movie Friend is the Web site developed as part of this research project. See Chapters 3 and 4 for description.

Figure 2-2: Technique utilization summary for research projects

2.3.2.2 Dynamically generating structure

Content-based filtering (CBF) and collaborative filtering (CF) are the two leading techniques used in generating adaptive Web sites. Much adaptive Web site research has focused on these filtering techniques used individually, used together, used in combination with data mining of server logs, and used in conjunction with user profiles.

CBF attempts to analyze files to determine their content and to return the set of files whose content matches the user's preferences. The text file example in section 2.4.2 reflects *syntactic* content checking. Note, however, that *semantic* content checking, as some systems attempt to do, has not been performed in the example. Both syntactic and semantic checking can be particularly difficult for files other than text files, since content is not always easily defined for these files (see Kohrs, et al., 1999).

Despite these limitations, CBF is employed in the Page Gather algorithm, a "statistical cluster mining component", which lies at the heart of the IndexFinder system (Perkowitz, et al., 2000). At the conceptual clustering level, the Page Gather algorithm makes decisions about the site's structure based on the content of files, and generates proposed index pages. While the system does dynamically generate structure for the site, the Web master must ultimately approve or deny proposed index pages in the IndexFinder system. For this reason, the IndexFinder system could be considered semi-automatically adaptive and not automatically adaptive. A similar approach is taken with the conceptual clustering algorithm COBWEB (Fisher, 1996). Like Page Gather, the COBWEB algorithm produces an index page of similar links based on content criteria.

Collaborative filtering (CF) relies on the maintenance of user ratings of items that comprise the Web site's data (files, links, pages). This is most often accomplished

through the maintenance of a user profile, which typically takes the form of a user ratings matrix. It is in terms of this ratings matrix that the problem of *sparsity* and the *first-rater* problem mentioned in section 2.2.4 are defined. Clearly, a user ratings matrix that is sparsely populated is of little value, and an item that has no ratings cannot be recommended.

This drawback has led most researchers to employ both CF and CBF in situations where CF is used. When no user ratings exist for an item, CBF is used until users have rated the item. Even so, some early adaptive systems, such as Ringo (Shardanand, et al., 1995), successfully employed apparently stand-alone collaborative filtering systems. The Ringo project brought forth the idea of a "social information filtering system", a Web-based music recommendation service where users could create a user profile, rate a series of 125 albums and then get recommendations (Shardanand, et al., 1995).

It's unclear, however, whether this is achieved without the assistance of CBF. While perhaps not considered CBF, any categorization or classification of music into genres for recommendation, implies content-based definition, and if that file information is used in the structure of the site then this could be viewed as content-based filtering. The key consideration for this type of structure is whether new items are added to the system, and if they are, how do they get their initial ratings or otherwise get recommended?

Extending the idea of a user profile, the AVANTI project (Fink, et al., 1998) added a *user model* as a key component of their adaptive system. The AVANTI project was undertaken to develop a hypertext-based, distributed information system that can be accessed at home, and at kiosks in metropolitan areas to give public information (local services, transportation, building locations) to visitors, tourists, citizens, etc.

At the core of this project was the consideration of the capabilities of the system's users (blind, wheel-chair bound, elderly, etc.) Since the system's targeted users exhibited varying capabilities, a clear goal was to provide user-oriented adaptivity. This led to the development of a user model and the classification of users. The user model is developed from an initial interview from which *primary assumptions* are derived. As the user performs actions, *inferences* are made based on these actions and *primary assumptions* (Fink, et al., 1998).

A powerful and unique approach, the AVANTI project showed some of the benefit of user models instead of CBF and CF in developing adaptive sites. However, its goal is based more on user class customization than on individual customization, and it presumably relied on a great deal of presumptions when developing classes and relegating users to them.

The AVANTI project did show just how powerful a tool the user model can be in generating adaptive sites. In fact many of the most successful commercial adaptive Web sites utilize user models in conjunction with filtering techniques (see Kobsa, Koenemann & Pohl, 2001). Amazon.com's clique-based filtering, user-ratings capabilities, and recommendations imply an underlying user model combined with CF, a model similar to that employed at CDNOW. Both of these sites report that greater than 55% of their business is from repeat customers, well above the average cited by Forrester Research at 35-40% (Tchong, 1998).

2.3.3 Combinatorial approaches

Most of the combinatorial approaches to adaptive Web site development focus on using content-based filtering (CBF) in conjunction with collaborative filtering (CF) in an

attempt to neutralize the shortcomings of both. This is meant to address the problem associated with pure CF methods of *sparsity* and the *first-rater* problem, and the problem of deriving content associated with pure CBF methods. As Balbanovic states, an approach that combines both CBF and CF can "incorporate the advantages of both methods whilst inheriting the disadvantages of neither" (Balbanovic, et al, 1997). This "hybrid content-based/collaborative system", called FAB, incorporates a user profile, and notes the development of this profile as key to the successful functioning of the system.

While much of the research doesn't strictly define a user model system component, most of them imply a user model, in terms of the system's assumptions about users based on the ratings they make. For example, these assumptions take the form of, "If user A likes movie B, and movie B is Science Fiction, then user A likes Science Fiction".

Polcicova echoes the advantages espoused by Balbanovic . This research shows that a combined approach (CF and CBF) can make up for the shortcomings of both (Polcicova, et al., 2000). For example, in cases where the content of an item is not known, CF can produce a recommendation for that item based on the user profile and the experiences of a similar user(s). In situations where ratings are missing for CF, CBF can recommend the item to the user based on content. This is successful when content or ratings information is available, but when neither is available, recommendation becomes difficult.

Another researcher to promote CF and CBF in combination is Melville, who proposed, "content-boosted collaborative filtering (CBCF)" (Melville, et al., 2001). In this research a key part of the approach involves a user ratings matrix we considered previously a user profile.

Perhaps one of the more intriguing adaptive research projects to use a combination of techniques is the Active WebMuseum (Kohrs, et al., 1999). It combines CBF and CF to generate user-adapting virtual corridors of an on-line museum, where users can browse and rate paintings, sculptures and artwork. The user can modify corridors of the museum by specifying artistic periods or styles, among other criteria. While a user model is not explicitly defined as part of this system, it is inferred as user preferences based on ratings, and user ratings are stored in a manner consistent with previous researcher's definition of user models. Kohrs (1999) sums up the benefits of a combined approach:

> In cases where collaborative filtering is limited by an insufficient amount of users and ratings, a combination of content-based and collaborative filtering should lead to better filtering performance. Besides the improvements of performance for cases of sparsity, a system which uses a combined approach can also recommend items which have not yet received any ratings e.g., new items, which is not possible for a system relying only on collaborative filtering. (p.33)

These combinatorial approaches provide the most promising leads in developing adaptive Web sites, at present, and with the reliance of each of these methods to some degree on user information, it's clear that the importance of the user profile should not be underestimated. Clearly, user models are a valid and promising place to start when exploring how to improve and extend these combinatorial methods.

CHAPTER 3

THE MOVIE FRIEND HYBRID COMBINATORIAL APPROACH

As discussed in the previous chapter, a key component in generating adaptive content for some of the research projects is either a well-defined user model, as in the AVANTI project (Fink, et al., 1998), or an implied user model (Balbanovic, et al, 1997; Kohrs, et al., 1999). Whether this takes the form of a user ratings matrix from which assumptions are derived about similar users, or exists as a defined user model, is unclear for most existing commercial systems that utilize these techniques.

For this reason, the Movie Friend Web site that is developed as a key part of the research of this project seeks to take advantage of the power provided by a clearly defined, but flexible user model. An explicitly defined user model, in conjunction with content-based and collaborative filtering, provides the foundation upon which the dynamic generation of the Web site's pages is achieved.

While much of the previous research into adaptive Web page development has focused on the monitoring of user access patterns and data-mining of server logs in conjunction with data filtering techniques, the Movie Friend project takes a different approach. Because of the prevalence of research in these areas, the Movie Friend Web site forgoes the oft-employed practices of monitoring user access and server logs. With its focus instead centering on the value of a well-defined, evolving user model and

assisted by the two primary data filtering techniques, the Movie Friend site gives a new perspective on the type of dynamic contribution a proper subset of techniques can provide to adaptive Web site generation efforts.

3.1 Evolving User Model

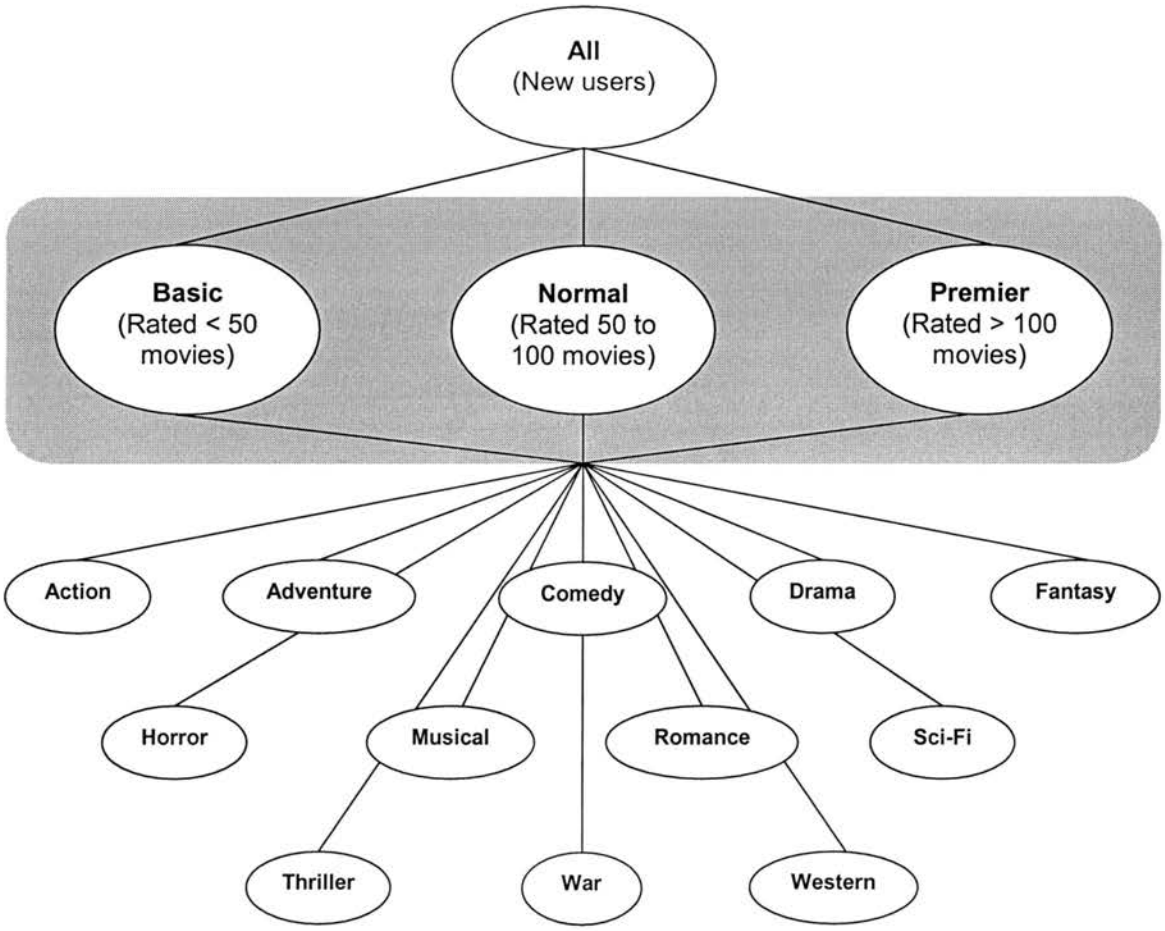
The user model for the Movie Friend Web site provides invaluable information to the recommender system that plays a primary role in the site's functionality. The well-defined user model exists as tables in the database (discussed further in Chapter 4 of this work) and allows for the categorization of users of the system. Through this categorization, similar user preferences are identified and recommendations to users with similar preferences are made.

Essentially these categorizations can be viewed as classes to which users belong. Initially a new user is placed into an *All* class since the system has yet to identify any preferences for the new user. In this class, no assumptions are made about the user as to preferences due to a lack of information about the user. This prevents the system from restricting or filtering content for a user based on the user model. User actions such as rating movies constitute actions for which the system can apply assumptions, and as these assumptions are made, criteria can be applied to verify them and the class(es) to which a user belongs may change.

The criteria applied here can be very extensive or complicated, involving techniques commonly associated with artificial intelligence and the development of heuristics in the best implementations. But for discussion, a simple example will suffice: For example, the system might set the criterion to be that a user must rate at least five movies 4 or higher on a scale of 1 to 5 (with five being the highest rating) in order to presume that the user

likes movies of that particular genre - let's say Science Fiction . When the criterion is met, the user is presumed to like Science Fiction movies. This information is added to the user model, our user is no longer in the *All* category, and the system can recommend the subset of the highest rated Science Fiction movies in the database that the user hasn't already rated. In this way the user model is evolving or dynamic, and the system relies heavily on the user model in producing the customized pages tailored to the specific current user.

A user model similar to the one discussed here is employed by the Movie Friend Web site and is reflected in the Movie Friend user model class diagram of Figure 3-1.



* Note that classes depicted in the shaded area are used as an integral part of the recommender system, but since these are easily determined at run-time and are potentially changing rapidly, they are not explicitly defined in the user model. All other classes depicted are explicitly defined.

Figure 3-1: Movie Friend user model class diagram

3.2 Data Filtering Contributions

Rounding out the combinatorial approach for the Movie Friend prototype are the essential contributions made by the two most prevalent data-filtering techniques presently used in generating adaptive Web pages. Without the addition of content-based filtering (CBF) and collaborative filtering (CF), the Movie Friend site cannot take advantage of the knowledge base that the evolving user model provides. Nor can it effectively disseminate the vast amount of data contained in the Movie Friend database to a widely varied audience with differing preferences and personal tastes.

In order to enable CBF, each movie entered into the Movie Friend database is placed into a *Movies* table containing the information related to the specific film (a unique movie id, title, year, director, synopsis, and ratings average). In addition, a second table, *Genres* (containing the corresponding unique movie id and a genre), maintains at least one row for each entered movie that corresponds to the genre the movie is in (action, adventure, etc...).

Many movies belong to more than one genre, and a row is created in the *Genres* table for each of these categories. It's only necessary that there be at least one corresponding entry in the *Genres* table for each movie in the database for the recommender system portion of the site to take advantage of the user model and for the system to be able to list all the movies of a particular type. However, the system works best when all the categories associated with a particular movie are entered into the system and a row for each genre a movie is a member of exists in the *Genres* table.

In addition, CBF is also utilized in the site's search facilities which allow users to specify keywords, such as a title, director or actor and get a listing of all of the movie

records that contain the specified keyword(s) in the title or director fields of the *Movies* table or in the name field in the entries of the *Actors* table. Since many actors may be associated with one movie, this *Actors* table is similar in structure to the *Genres* table, containing fields for a unique movie id and an actor name.

CF is used in virtually all of the functionality the site provides. All keyword search results, genre listings, listings of personal favorites and the site top ten are produced utilizing the ratings averages stored in the constantly updated average field for each title in the *Movies* table of the Movie Friend database. As users rate additional movies, as users seek recommendations, or elect to list the top ten movies at the site, the ratings averages for all movies at the site are re-calculated based on all user ratings and stored in the database. Each user-initiated listing at the site is then produced in descending order, such that the highest rated movies are at the top of each listing, down to the lowest rated movies at the bottom.

While the user model assists CBF to a great extent, it significantly empowers CF with the ability to more effectively produce results from the database that are not only the results of examining the ratings matrices for a community of users, but also correspond to the user preferences and attributes provided in the user model. This is most apparent in the recommendation features of the system and is where the power of CF is most effectively exploited.

Clearly, CF benefits greatly from the user model, but without CBF, much of its power is lost. In effect, CBF allows us to identify the nature of the data at the site and CF allows us to identify the opinions, preferences, and desires of the users who seek that data. Together with a better understanding of the system's users, in terms of the user model,

this provides a powerful combinatorial approach upon which the Movie Friend prototype Web site is built.

CHAPTER 4

MOVIE FRIEND PROTOTYPE IMPLEMENTATION

While any Web site imaginable could benefit from the combinatorial method discussed in the previous chapter, prior to development, the decision is made that the prototype for this research project should not only take advantage of all of the potential of adaptive Web site generation techniques, but should demonstrate in the clearest possible way the adaptive functionality it possesses to aid recognition and value assessment.

Secondly, the recommender system for an adaptive site that utilizes these techniques relies heavily on collaborative filtering of data based on a community of users. For this reason, the system strongly depends on the user bringing a knowledge base of their own about the site's data. In essence, for the system to be effective, it must trust the opinions of a group of users that are fairly well educated about the site's content.

Lastly, the system should provide the system developer, users and testers with as high a level of enjoyment and entertainment as possible. This is intended to facilitate completion of the system's development and to promote system use by users and testers.

For all of these reasons, the Movie Friend prototype Web site developed focuses on movies, and provides a community of users with the ability to peruse the site's database of movies, searching for favorite titles, actors or directors, and listing movies by genre, as one might browse the local video store. Unlike the local video store; however, users have

the added capabilities of rating movies, listing their favorites and getting recommendations based on other users with similar tastes.

This type of system facilitates exploration of the combinatorial approach at the core of the system in a fundamental and instantly recognizable way – rating movies in one fashion or another is a widely accepted practice, from newspaper articles to the local TV guide, to nationally syndicated movie critic shows.

Watching movies is a favorite pastime for millions of people around the world; thus, movie fans provide the system with an almost infinitely broad user group that is well educated about the site's data. And, as indicated by the commercial success of the movie industry, the entertainment medium is clearly recognized and enjoyed by millions of viewers.

4.1 Development Decisions

An adaptive system need not be inherently Web based, as discussed in Chapter 6 of this work. In fact, an intriguing aspect for the future course of development in adaptive systems may indeed be the exploration of adaptive applications.

For this project, however, the focus is the Movie Friend combinatorial approach to generating adaptive Web pages. As such, the project's intent and scope lends itself to a Web-based approach and influences the tools used for implementation. These considerations play a key role in initial development decisions made as part of the project.

4.1.1 Web-based implementation

The focus of this project specifically centers around methods and techniques used to develop adaptive Web sites. Obviously, for this reason, a Web-based implementation of

the Movie Friend Web site is inherent in the development of this project. Nevertheless, other valid reasons exist for the Movie Friend prototype's Web-based implementation.

The availability and prevalence of the existing software for utilizing the developed system weighs heavily in development decisions for such a system. With the widespread availability of "free" Web browsers and the prevalence of browsers packaged with the predominant operating systems used on personal computers today, a Web-based implementation makes sense.

Also, since the variety and type of systems the users will be using to access the system is unknown, a Web-based implementation avoids the "platform" compatibility issues typically associated with development of a commercial software product. All that needs to be done in development is to test the site with the various common browsers the system seeks to support, without regard to the end-user's operating system or system hardware configuration.

Extending this cross-platform compatibility, the system is developed within the "client-server" network computing model, taking advantage of server-side scripting to provide its functionality and content. This allows all script execution to take place on the server as opposed to other approaches where application execution may take place on the client's machine. This has the drawback of increasing more of the processing load on the server, but allows for maintenance, and observation of the site's performance, without much consideration for the limitations of the client's machine.

The site's content also lends itself to a Web-based presentation. Other complimentary Web sites to the Movie Friend site, such as the Internet Movie Database

(www.imdb.com) are already providing content of a similar nature on the Web, and should be familiar to many users of the Movie Friend system.

Furthermore, the ever-evolving user model at the core of the Movie Friend system, the constantly updating database and the site presentation based on a community of users' collaboration are all more easily dealt with and implemented with a Web-based design.

4.1.2 Tool selection

The Web-based implementation of the Movie Friend site is the primary consideration when evaluating the development tools, programming languages and software that make up the components of the system. HTML, as the de-facto standard language for creating pages on the Web, is the primary language needed for formatting and displaying the site's pages. Still, HTML only gives the development effort a piece of the puzzle needed to implement the site's functionality and dynamic display of data.

For this reason, a second language is needed to compliment the HTML that makes up the site's pages, providing the functional characteristics of the site and generating the dynamic content.

Active Server Pages, a proprietary language from Microsoft Corp., provides the dynamic capabilities sought and is available for free; however, it must be installed on a Web server running Microsoft's Windows 2000, Windows 2000 Server, or Windows XP. The Web-based system would then have to interact with a Microsoft Access database, part of another Microsoft software package (Microsoft Office 2000/XP), to access, store and update the data that makes up the site's content. None of these additional software packages are available for this project for free.

Given the facilities available to the development effort, this would require the purchasing and installation of machines and software dedicated solely to project development and would require the configuration and maintenance of a stand-alone server for the project. Fortunately, another, more attractive solution is available to the development effort.

Utilizing existing facilities available for project development, consisting of a student account on the Computer Science departmental Web server at Southwest Texas State University, project development is more readily undertaken. Running Linux and served up by an Apache Web Server, the departmental Web server allows for more cost-effective implementation of the Movie Friend project and allows the development effort to focus on the key research in the project.

Together with the use of a student MySQL database account, the student Web server account provides the key components that are at the core of the Web site implementation. Rounding out the solution is the use of PHP, an open-source scripting language, which allows the type of functionality and dynamic content generation needed in the development effort.

This approach has the added benefit of eliminating the need for administrative access on the Web server. Since this project doesn't seek to discover user access patterns at the site, nor to monitor server logs, this has no project drawbacks. Additionally, server installation, configuration and maintenance, all routine tasks that have no bearing on the research in this project, are avoided.

Despite the monetary savings, as an added benefit, developers get the peace of mind in knowing that their work supports the open-source philosophy among developers in the

software community, seeking to develop, debug and test software through a collaborative effort amongst professionals in a world-wide community. Since the software is freely distributed, the focus is on producing truly functional, usable software, avoiding the profit-driven goals of multi-billion dollar software companies.

4.2 Prototype Design

The design of the Movie Friend prototype is driven by a desire to make the site as user-friendly as possible, and to make the site visually pleasing, primarily because of the nature of the content at the site. This content, devoted to movies - in large part a visual entertainment medium – provides a great opportunity to take advantage of graphics and thematic display that is stimulating to users. For these reasons, significant attention to human factors in system design is given during development of the site's pages.

Also, a clear aspect of the site's functionality from initial development is that the site must provide a mechanism for "secure transactions". While no monetary transactions actually occur on the site, as no commerce is intended to take place, a mechanism for secure log-in is necessary, so the system can identify the user for tasks such as displaying the user's favorites and generating the user's recommendations. Indeed a very fundamental requirement of the collaborative filtering methods and of the evolving user model is the need to identify the user.

This user authentication is further needed to protect users' identities and passwords to prevent intrusion of user accounts and corruption of user data. This user identification has the added benefit of allowing the administrators of the site to observe system users to verify that their behavior is "appropriate" within the user community. For example,

system administrators would clearly want to be aware of behavior at the site that adversely affects the site's content.

If a rogue user runs rampant at a site, rating all movies at the lowest rating, administrators would want to know this to make the appropriate corrections to the site's database. Likewise, if a user posts something offensive or inappropriate, the administrators would likely wish to remove it and possibly de-activate that user's account. Obviously, user authentication is a major concern for any site that allows users to alter the site's data, or post new data to the site.

4.2.1 Web site architecture

The Web site architecture for the Movie Friend system is determined primarily by its varied functionality, which, from a design perspective, allows the division of the system's pages into two basic categories – protected and unprotected. Some of the system's features, such as searching for movies by title, actor or director, listing the site's top ten, and listing movies by genre can be and should be provided to anonymous users.

There is no reason to protect these actions on the site since they do not involve any user-initiated alteration of the system's data. Furthermore, some functionality should be provided without requiring users to log-in or create accounts. This is intended to attract more users to the system, giving them a "taste" of what is available at the site.

However, as mentioned previously, some of the more powerful features of the site, such as tracking and listing user favorites and generating recommendations, require user authentication, simply for the purpose of identifying the user. While it is not necessary to protect pages that allow these actions, identifying the user is necessary. And since the

user is known, the nature of "session control" within the scripting language PHP simplifies the design if these pages are incorporated into the protected areas of the site.

Accordingly, actions such as rating and reviewing movies involve the alteration of the site's database, and thus, must be protected. These considerations lead to the Movie Friend Web site architecture that includes both protected and unprotected pages, which is depicted in Figure 4-1.

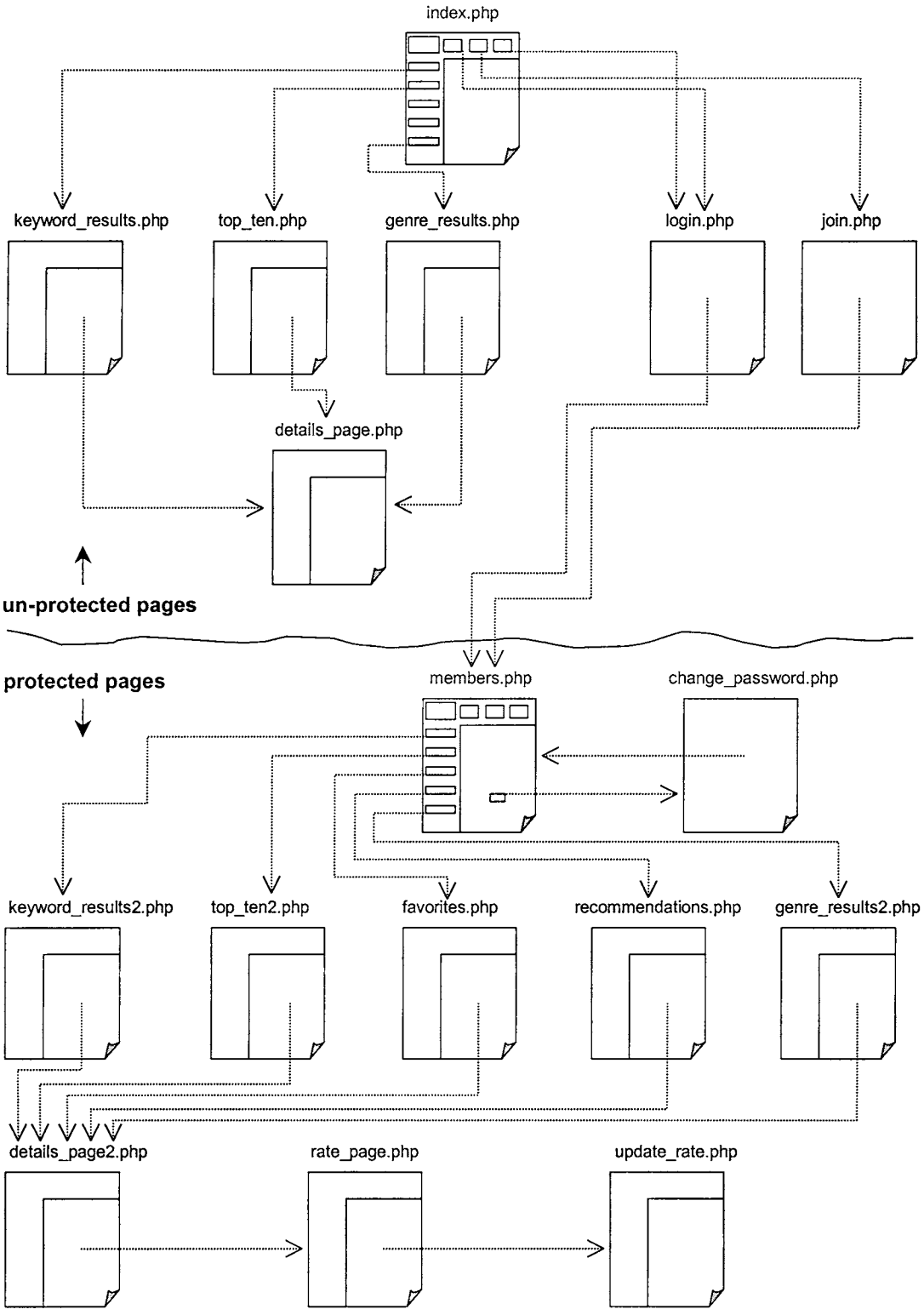


Figure 4-1: Movie Friend Web site architecture

4.2.1.1 Unprotected pages

As shown in Figure 4-1, the initial entry point for the Movie Friend Web site is from the page generated by the script *index.php.* This page is displayed when the user points his or her browser to the URL *http://www.cs.swt.edu/~shane/moviefriend* and is the home page for the site (See Figure 4-2, later in this section for the Movie Friend home page).

Obviously, the user, once aware of the files in the site's structure, can access the site through multiple entry points, by typing in the full path to the page in the browser's location bar. This is no concern to site administration, however, as using any of the unprotected pages as an entry point results in no different behavior than normal "click-through" access of the site. And since the protected pages' content is accessed within a PHP session, only users who have logged in and been authenticated by the system can access the dynamic content presented on the pages within the protected realm or perform the actions within these pages that alter the site's database.

In Figure 4-1, we see that the pages that provide the content to anonymous users discussed in the first paragraph of section 4.2.1 are implemented as unprotected pages for the site. These pages (generated by the scripts, *keyword_results.php, top_ten.php, genre_results.php,* and *details_page.php*), are accompanied by two additional unprotected pages generated by the scripts *login.php* and *join.php*.

The script *login.php* generates a page from which the user can enter a username and password to access an existing account, while the script *join.php* generates a page where the user can create an account on the system, entering a username and password and then verifying that password. These two pages must reside within the unprotected pages of the site to allow new and returning users access to the protected pages of the site. Both of

these scripts forward the user to the protected members page of the site, generated by *members.php* upon successful log-in or account creation.

4.2.1.2 Protected pages

From the members page, the user can access the protected pages at the site, taking advantage of the additional features that provide customized data presentation for the individual user. The listing of user favorites (if the user has rated any movies) is delivered in the page generated by the script *favorites.php*. Individual user recommendations are produced on the page generated by the *recommendations.php* script – again, depending on the number of movies rated by the user and other criteria discussed later in this work.

Note that from Figure 4-1, there are several pages generated by scripts with very similar names to those in the unprotected group of pages. The *keyword_results2.php*, *top_ten2.php*, and *genre_results2.php* scripts produce the same page output as their unprotected kin. However, they differ only in that selecting the "Details" button for a movie title on these pages links to the protected page generated by the *details_page2.php* script. This is necessary in the system design because the protected details page contains a "Rate It" button that links to the page generated by the *rate_page.php* script. Clearly, only those users who have been identified and authenticated, and are within the protected pages of the site, should be able to rate movies or provide reviews.

The page generated by the *update_rate.php* script is displayed after the user selects the "Submit" button on the rate page. It notifies the user of a submission error or shows the user's just-submitted rating and review and a message acknowledging success in adding the user's rating/review submission.

The final page that makes up the group of protected pages at the site is the page generated by the *change_password.php* script. It is decided in development that the ability to change a password is most easily implemented in this architecture as a protected page within the PHP session. This is because in order the change a password the user must first provide his or her username and old password, for verification purposes. Otherwise, a user aware of another user's username could just provide the username and change that user's password.

In essence, the user must first log in under their old password to change it to a new one. Obviously it would be redundant to ask a user who is logged in for their username and old password, since the system already has that information! For this reason the determination is made that the change password functionality should only be available from the members page, where a user is already logged in and authenticated into a protected PHP session. Then the *change_password.php* script need only generate a page that displays the current user's username in an un-editable text field and requires the user to enter the new password twice.

Additional files and PHP scripts, beyond those that generate the pages shown in Figure 4-1, exist in the Movie Friend implementation. See Figure 4-7 for a listing of the 23 files that comprise the Movie Friend Web application and a brief description of their purpose and functionality.

4.2.2 Page design

During project development, user considerations standard to all software development efforts are followed and implemented in the design of each of the site's pages. These include such things as considering the target user group, making sure that the site's pages

provide the appropriate responses to navigation actions at the site, ensuring that the functionality of the site is readily understood by a varied user group, and that this functionality, when activated, provides appropriate and meaningful responses to the user. This is all undertaken with the intent of achieving the underlying goal of a high degree of usability for the user and to enhance user experience with the site.

The user responses incorporated into the site's pages include the use of JavaScript for all of the page headers and sidebars, which are available on each page of the site. JavaScript provides the capabilities of providing changing background colors when a user uses the mouse to pass over an area where a link to another page or functionality exists. Referred to in JavaScript programming as a "mouseover", this visual indication allows the user to recognize areas of the page where a page or functional link exists prior to selecting that link.

In essence, this provides "hot" portions of the page where the functional aspects of the page are easily discernable from the non-functional ones. Furthermore, since it is a commonly used technique on pages throughout the Web, it allows the user to bring some of his or her own experience and familiarity with Web sites to page navigation through the Movie Friend Web site.

Contributing to the response methods of the site are the built-in features of the browsers used to access the site. For example, the changing of a mouse pointer to a hand when a link on the page is passed over, and providing audible "clicks" when a selection is made.

Additionally, the site uses HTML forms to handle user selections, data entry and data submission. This is done primarily to take advantage of the built-in features of HTML

forms, such as the highlighting of user selections in selection boxes, and visual feedback from button selections in terms of the appearance that the button is "depressed". The added benefits of user familiarity with HTML forms and site consistency where data is collected from the user or the user interacts with the site, are also guiding factors. (See Figures 4-2 through 4-5 for a sampling of the pages that comprise the Movie Friend prototype Web site).

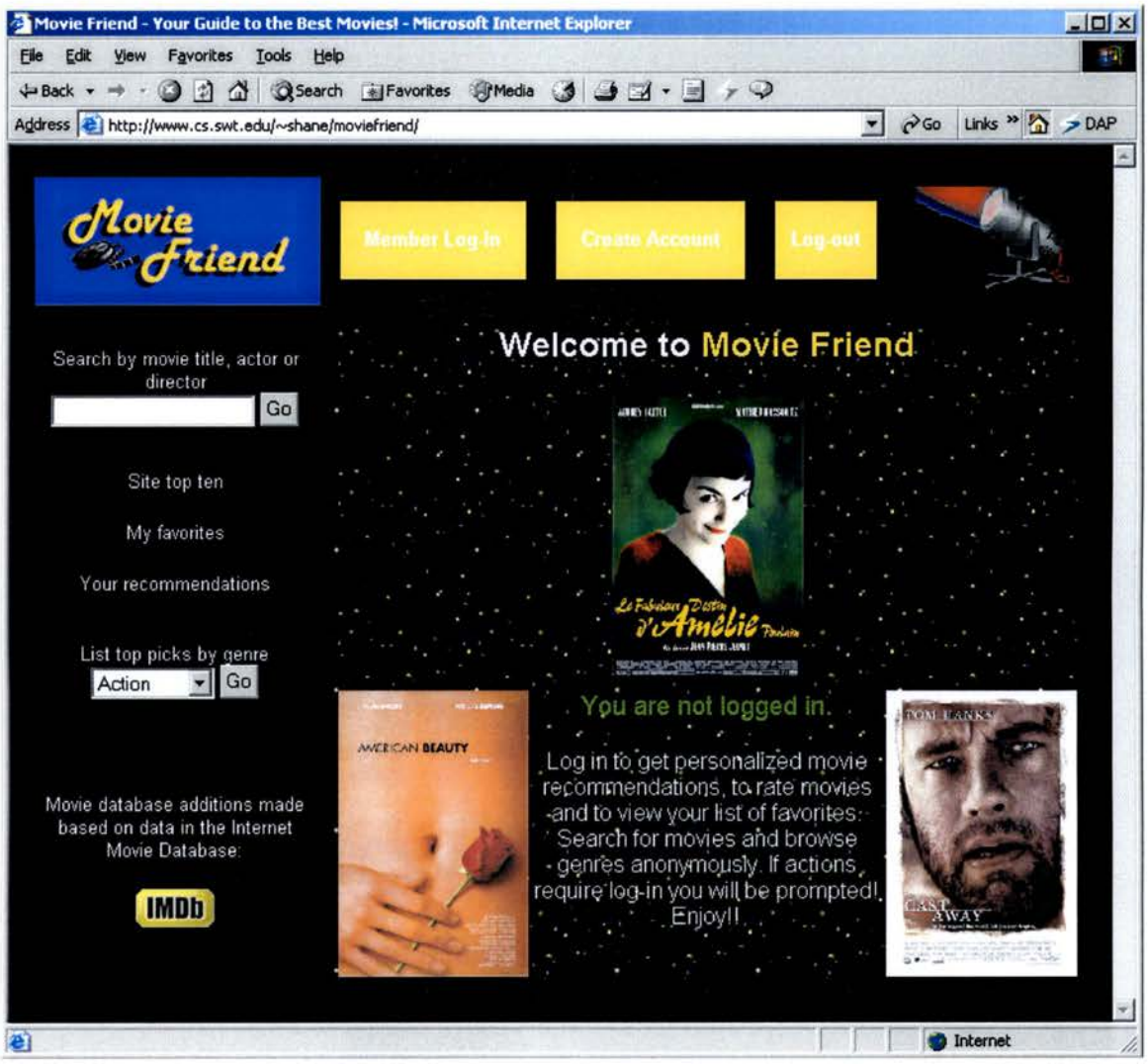


Figure 4-2: The Movie Friend home page, *index.php*

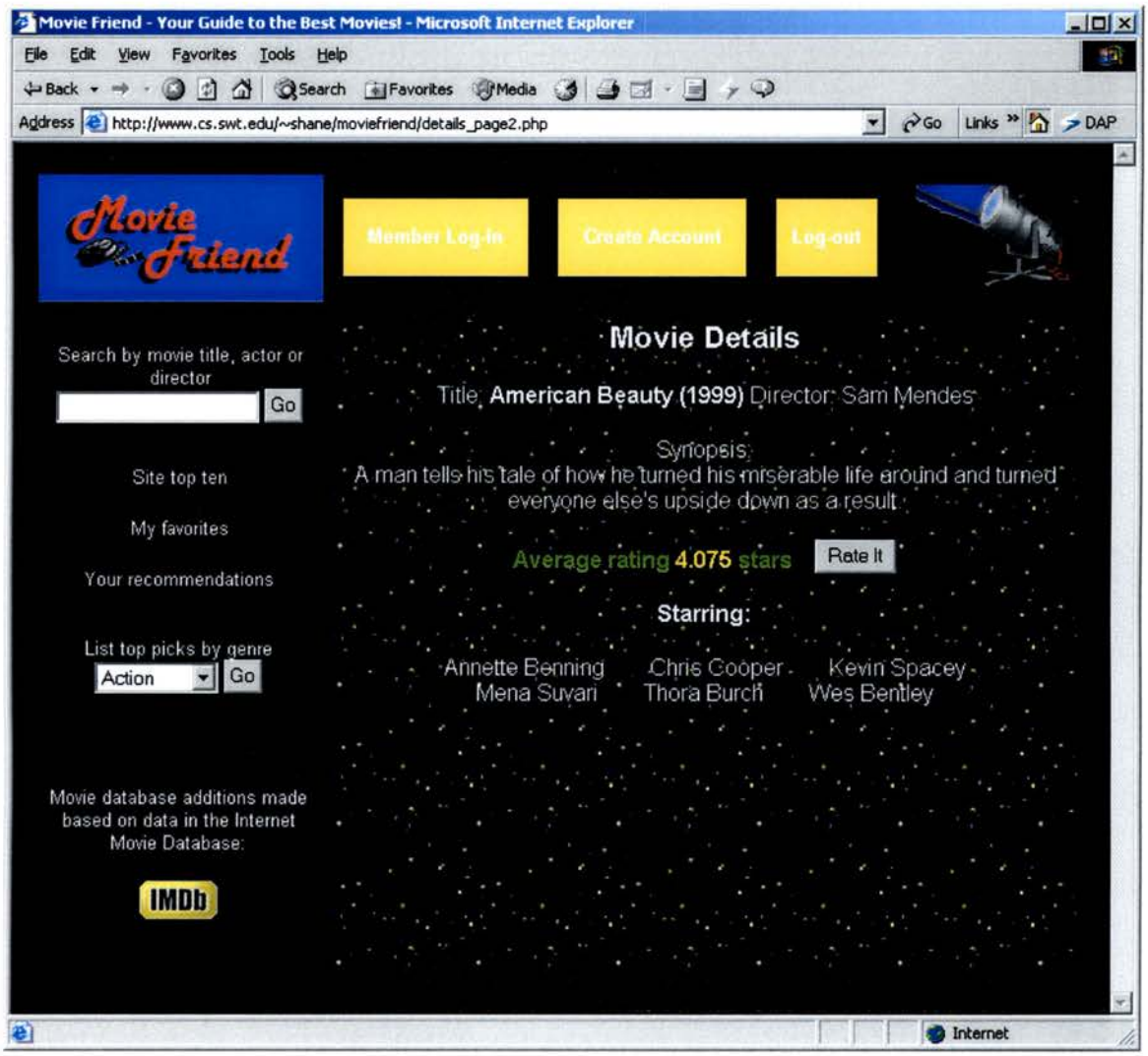


Figure 4-3: The Movie Friend Movie Details page, *details_page2.php*

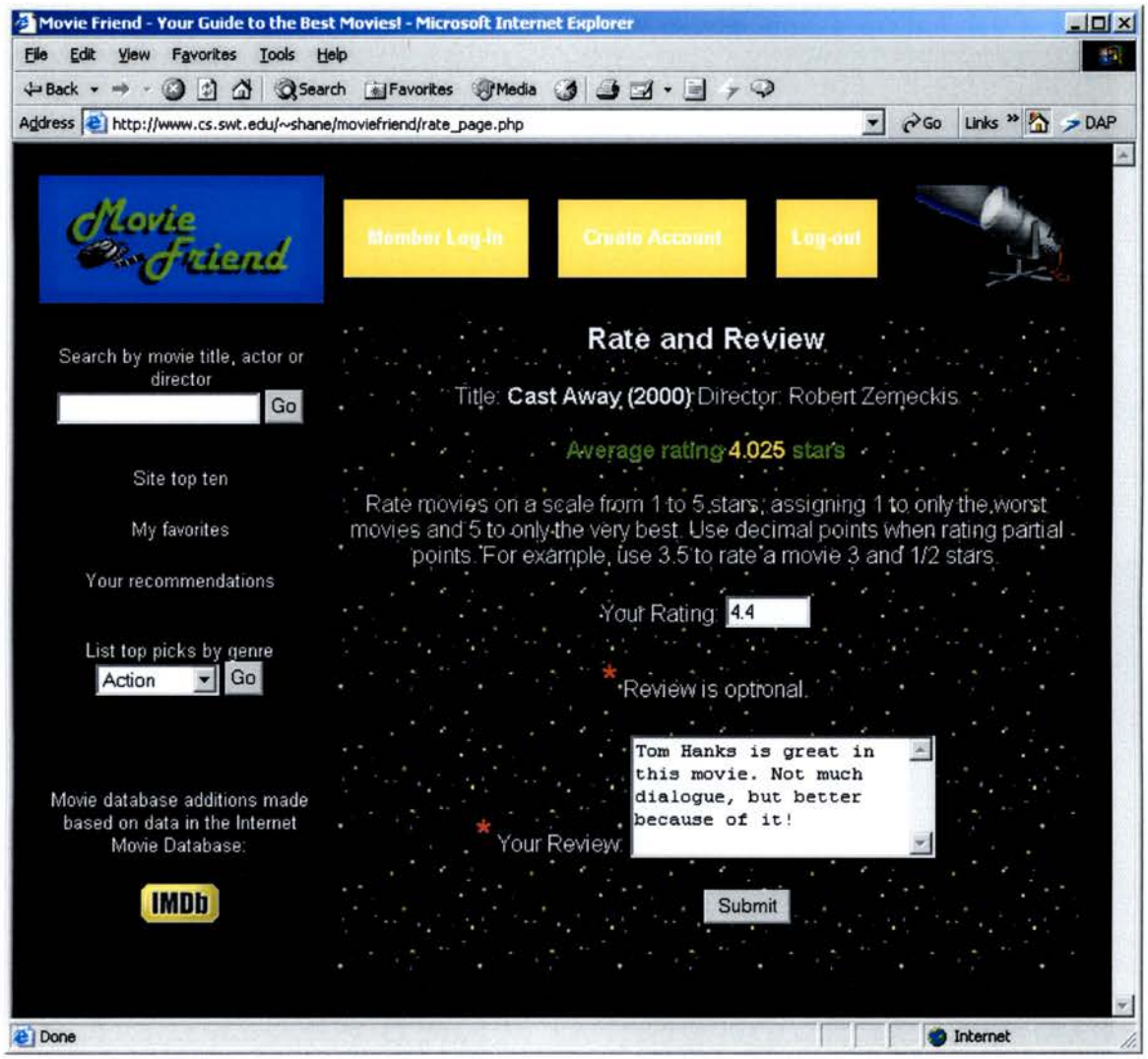


Figure 4-4: The Movie Friend Rate and Review page, *rate_page.php*

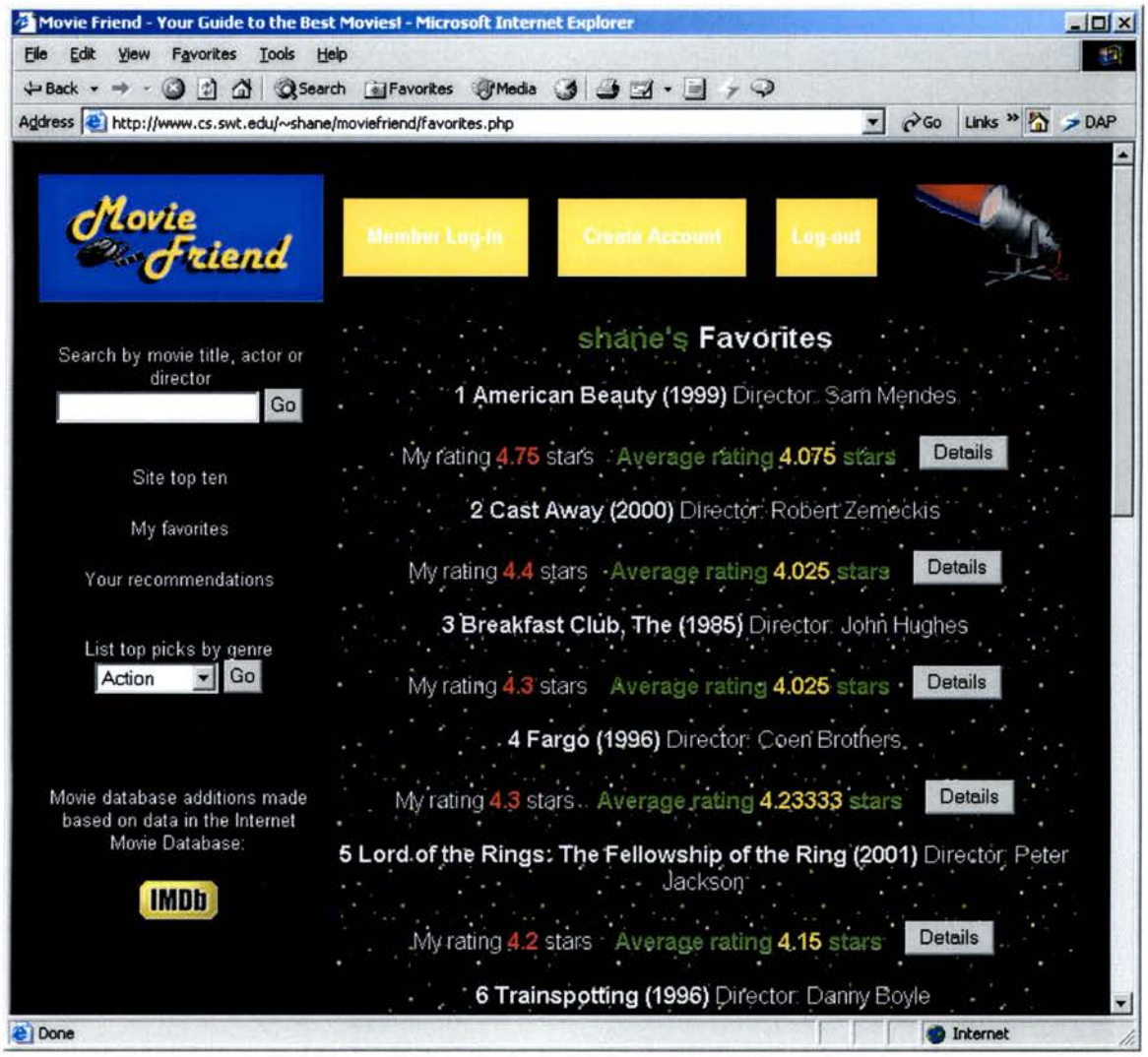


Figure 4-5: The Movie Friend Favorites page, *favorites.php*

As mentioned above, a main consideration for the site's pages is consistency of presentation, both visually and thematically, and to aid the user's navigation through the site's pages. This is achieved by using the files *header.inc* and either *sidebar.inc* or *sidebar2.inc* as part of all the site's pages (*sidebar.inc* is used for the unprotected pages at the site and *sidebar2.inc* is used for the protected pages).

It may not be immediately clear from the file names just what these files are. Basically, in PHP parlance, these are termed *include* files. They actually are ordinary HTML files, but by naming them with the *.inc* extension, system programmers are reminded of their function in the system. And since we want each page to have both the header and sidebar (for consistency, uniformity and navigation), it's only necessary to include these two files at the beginning of each additional PHP script created to generate additional pages. (See Figure 4-7 later in this section for more information about the files that comprise the Movie Friend Web site).

4.3 Web Site Scope and Intended Audience

The Movie Friend prototype provides a fully functional example of an adaptive Web site that utilizes user models, content-based filtering and collaborative filtering to automatically generate its pages. This prototype Web application provides the groundwork for testing the feasibility of generating dynamic Web pages based on these three key techniques. It does not seek to provide a comprehensive database from which users can gather information about the hundreds of thousands of movies that have been produced over the last 100 years or so.

While the site's baseline data is based on the content contained at the Internet Movie Database Web site (*www.imdb.com*), there is no implied or intended partnership between

this effort and the commercial offerings of IMDB. Content licensing proves too cost-restrictive for this research effort, therefore there is no connection between IMDB's movie database and the Movie Friend database, although the developer thankfully acknowledges the use of baseline data attained from IMDB for the Movie Friend prototype's research database. This database features 100 movie titles, only a very small percentage of those listed at IMDB, but a number deemed sufficient to test the focus of this research effort.

Additionally, a significant portion of the programming for the prototype centers on the recommendation features, since this is where the power of the user model is most greatly exploited. Because of this, it is determined that the prototype would benefit most from having a database comprised of records of many of the more popular (and award-winning) movies of the last century, while attempting to get an appropriate number of representatives from each genre, and adding a number of the author's favorite movies for good measure.

The project's intended audience further justifies the decision to populate the prototype's database with well-known movies. While the audience for a similar commercial site might seek to provide content for a community of millions of users, the intent of the prototype is to provide content for a small user group for testing purposes. With the prototype's comparatively small number of movie titles and related information, testing likely benefits most from using a database of movies that are recognized by the largest number of users in the test group.

4.4 Key Functional Components

The PHP scripts developed for the site provide the primary functionality for the system, interacting with and updating the prototype's database, and producing the dynamic content presented on the site's pages. These scripts consist of PHP code that processes user actions, interspersed with embedded HTML for page formatting and display, and HTML forms for accepting user data input. Together with PHP, the form data can be processed and entered into the prototype's database as necessary or used to set criteria for querying the database and returning results.

The flexibility of PHP allows for a great deal of co-mingling of languages, as PHP code can be embedded within HTML or HTML embedded within PHP at will, as long as the PHP interpreter is installed and running on the Web server. Accordingly, a set of pre-defined PHP functions is available that allow connection to and interaction with the MySQL database, among other capabilities. These activities exist simply as function calls within any script and can be used freely within a PHP code block.

To query the database, database connection variables are first established in a configuration file for the project, *config.php*, and a function called from this file connects to the MySQL database. Once this is established, each of the scripts that query the database need only include the configuration file to connect to the database.

4.4.1 MySQL database

A great deal of care is taken when designing the database for the Movie Friend prototype Web application because of the problems associated with an improperly designed database. These problems range from data redundancy, to performance issues

when performing actions on the database, to data anomalies caused by improper database structure.

Data redundancy occurs when differing database tables have too many of the same fields within them. This directly relates to performance or storage issues in that the database must store more data than is necessary. For a large database, this could mean a serious storage requirement for redundant data.

Data anomalies are also most often caused by these redundant fields or by attempting to store multiple values within single fields of a database table. For this reason, the database design reproduces only foreign keys in successive tables, and data stored in each field of a corresponding row in the database table is an atomic (single) value.

The tables in the Movie Friend database are depicted in Figure 4-6.

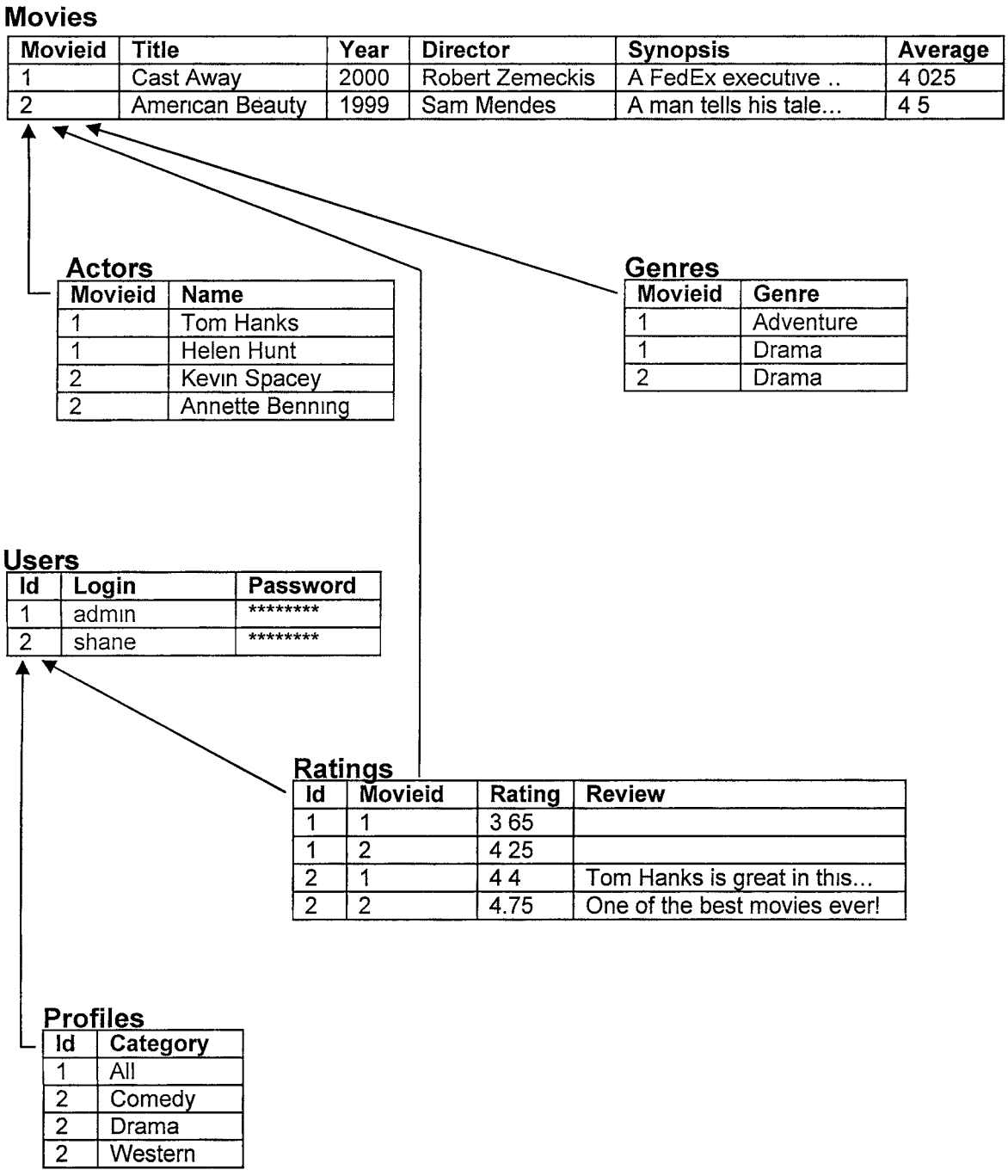


Figure 4-6: Movie Friend MySQL database tables

The tables are designed to only duplicate fields required to differentiate the additional data from one table to the appropriate row of another. The arrows that point from a field in one table to another indicate a *foreign key*. For example, the arrow pointing *from* the *Movieid* field of the *Genres* table *to* the *Movieid* field of the *Movies* table indicates that this is a foreign key for the *Genres* table.

The *primary key* for the *Movies* table is the *Movieid*, so only one row with this number should exist in the table. Conversely, the primary key for the *Genres* table is defined as the *Movieid*, *Genres* fields combination, which allows the database to maintain multiple rows with the same *Movieid* number, but differing *Genres*. This allows our database to maintain the proper structure discussed previously – avoiding redundancy, using atomic values for fields and maintaining data integrity.

For the remaining tables that have arrows pointing *from* a field *to* a field in another table, this same structure holds true – these are *foreign keys* in the first-mentioned table. In these cases, as in the Genres table, a second field in the table is used to create a *primary key* pair. As before, this allows for multiplicity of the first field in the table, yet maintains unique pairs within rows of the table and, hence, maintains the proper database structure.

4.4.2 PHP scripting

Numerous PHP scripts are developed as part of the project, not only to generate the Movie Friend site's pages, but also to implement the combinatorial approach central to this research project and to provide the site's dynamic content. These files include those depicted in the Movie Friend Web site architecture diagram of Figure 4-1, which are discussed in section 4.2.1 of this work, and the configuration file, *config.php*, mentioned

in section 4.4. While not responsible for the generation of entire Web pages at the site, two additional files not previously discussed provide key functionality for the Movie Friend site, and warrant brief consideration.

The most important of these additional files is *functions.php*, without which the project would be little beyond a static Web site. Adhering to modular design techniques, this file is developed to provide numerous valuable functions to the other scripts, which allows the source code of the scripts responsible for generating pages to be much cleaner in design and easier to read and evaluate during debugging.

Functions in this file handle everything from connecting to the database, adding user accounts to the database, initiating and terminating PHP sessions, verifying text entry for form fields, and displaying data on pages in a variety of ways, among other things. It also has the distinction of being the largest file at the site, in terms of lines of code.

The last file used at the site, *logout.php*, does not produce any pages or output. It is a simple script that uses a function to un-register the PHP system variables for a session and forwards the user to the unprotected home page for the site, *index.php*. Figure 4-7 below shows a complete listing and brief description of the 23 files that comprise the Movie Friend Web site.

File name	Type	Displays	Dynamic Functionality
header.inc	HTML with JavaScript	Page header for all pages	None
sidebar.inc	HTML with JavaScript	Unprotected page sidebar	None
sidebar2.inc	HTML with JavaScript	Protected page sidebar	None
change_password.php	PHP, HTML	Change password page	Updates user password in system database
config.php	PHP	None	Registers PHP session variables / connects to MySQL database
details_page.php	PHP, HTML	Unprotected details listing	Queries the database and returns information for the selected movie
details_page2.php	PHP, HTML	Protected details listing	Queries the database and returns information for the selected movie
favorites.php	PHP, HTML	Favorites listing	Queries the database and returns the user's highest rated movies
functions.php	PHP, HTML	Varied page content	Various functionality used by other scripts in the site
genre_results.php	PHP, HTML	Unprotected genre listing	Queries the database and returns all movies in the selected genre
genre_results2.php	PHP, HTML	Protected genre listing	Queries the database and returns all movies in the selected genre
index.php	PHP, HTML	Site home page	None
join.php	PHP, HTML	Create account page	Creates new user account and stores username and password
keyword_results.php	PHP, HTML	Unprotected keyword search results listing	Queries the database and returns all movies with title, actor or director keyword match
keyword_results2.php	PHP, HTML	Protected keyword search results listing	Queries the database and returns all movies with title, actor or director keyword match
login.php	PHP, HTML	Member log-in page	Authenticates user for system access and forwards them to the members page
logout.php	PHP	None	Un-registers PHP system variables and forwards user to index.php
members.php	PHP, HTML	Protected members page	None
rate_page.php	PHP, HTML	Rate and review page	Queries the database and returns rating and review for the current user and selected movie
recommendations.php	PHP, HTML	User recommend-ations page	Queries the database and returns movie listing based on user model and data-filtering methods
top_ten.php	PHP, HTML	Unprotected top ten listing	Queries the database and returns the top ten rated movies
top_ten2.php	PHP, HTML	Protected top ten listing	Queries the database and returns the top ten rated movies
update_rate.php	PHP, HTML	Information added page	Updates or adds user review and rating in system database

Figure 4-7: The 23 files that comprise the Movie Friend Web site.

4.5 Data Filtering Problems Addressed

The contributions made by data filtering provide a major component of the Movie Friend site that enables much of its functionality. Special consideration is given to the problems related to the two types of data filtering (discussed in Chapter 2 of this work) during site development, to ensure that these problems do not in some way affect the research effort.

As noted previously, a key requirement for effective content based filtering (CBF) is that the content of the site's data be either clearly defined through some additional mechanism, or clearly discernable from the data itself. This is much more difficult for multimedia data, but in the case of the Movie Friend site's data, we are dealing with data records made up of text fields and text description. This simplifies the gathering of content information on these records to searching for matching patterns in the fields of the database.

For example, the content of a movie record - in terms of the title, director or actor name - is explicitly defined by the text fields that make up the entries in the database. If the system needs to find all movies with Tom Hanks listed as actor or director, it need only query the database for the pattern "Tom Hanks". In essence, the actor name describes the content of the movie record in this example.

CBF is further assisted by the explicit definition of genres for each movie title that is entered into the database. This equates to another explicit definition of content and is performed to extend CBF and assist the second filtering method, collaborative filtering (CF) in performing some of its tasks.

Two additional problems associated with CF are recognized and effectively addressed in the Movie Friend prototype. The problem of *sparsity* and the *first-rater* problem, first discussed in Chapter 2, provide little detriment to this adaptive system because of the way data records are entered into the system.

The problem of *sparsity* is essentially a lack of user ratings for multiple items, defined in terms of a user ratings matrix that is sparse. This problem is nullified to a great extent, but is one of the trickier aspects of CF, especially in regard to the functioning of the recommender system which seeks to match similar users to similarly-liked data. However, the *first-rater* problem, which relates the necessity that an item must be rated to be recommended, is completely negated.

Fortunately, the development effort has at its disposal a comprehensive database of movie titles that includes, in many cases, thousands of user ratings for individual titles. By utilizing the existing user ratings information from the IMDB database for titles entered into the Movie Friend database as a baseline, the *first-rater* problem is eliminated and the problem of *sparsity* is minimized.

It is obvious how the *first-rater* problem is eliminated with this method. What may not be obvious is how the problem of *sparsity* is minimized. The answer lies in how the data is entered into the database. For the Movie Friend system, this is accomplished by creating a so-called "expert" user, called "admin" in the case of the Movie Friend prototype, which uses the ratings from the IMDB site as its ratings. For every movie entered into the database, there is a corresponding rating for the user "admin". This user's ratings matrix is anything but sparse - rather it is completely full.

Using this method, *sparsity* has less of an effect on the system. However, the system still has only one, albeit "expert" user, and CF benefits most from a community of users. For this reason it can be said that *sparsity* is not fully addressed in the initial implementation of the system. Fortunately, as more and more users join and use the system over time, the effects of *sparsity* on the system diminish.

4.6 Incorporated Features

As a result of the content the Movie Friend site delivers (movie data records) and some of the functionality it provides in conjunction with those records (specifically, the user's ability to add ratings to those records) the prototype exhibits a from of "pure" *promotion* and *demotion* within its pages, concepts first mentioned in section 2.3.2.1 of this work in relation to adaptive Web site design (Perkowitz, et al., 1997).

Effectively, items are promoted or demoted based on a constantly updated popularity score such that the more popular items gravitate towards the top of index listings, pages, etc. Since ratings averages are maintained for all movie titles at the site, these "popularity scores" are explicit within the data definition and not implied. Constantly updated as users rate movies, get recommendations or list the top ten, these ratings provide the "pure" *promotion* and *demotion* mentioned above.

CHAPTER 5

QUALITATIVE TESTING RESULTS

During the development phase, numerous tests were undertaken as part of the Movie Friend project to verify that the project functionality was achieved and accomplished in an efficient way. These tests include component-level testing of each of the scripts to ensure that, as functionality is added, boundary cases and error conditions are tested for each new process and that each functionality delivers results in a meaningful and familiar way to the user.

Adherence to standard programming techniques, establishment of database connections, and the tasks of updating and adding information to the database are all tested for success or failure to ensure data integrity, and data collected from page form fields is verified to be of the proper format for entry into the database. Comprehensive testing during the development phase is conducted to ensure that these system safeguards indeed function properly. This testing, while essential to the success of the project, is not discussed further in this work, rather the focus is on the qualitative testing that is conducted.

Whereas the previous testing discussed focuses on the developer testing the system during development, the qualitative testing that is the focus of this section concerns the tests conducted by the test user group for the project after the prototype has been fully

implemented. The choice to use a qualitative evaluation of the system is based on two factors:

First, the previously established system goals are not readily evaluated through some sort of mathematical or scientific evaluation. User-friendliness, ease-of-use, enjoyability, and degree of entertainment are all matters of user opinion and thus cannot be quantified in a meaningful manner for rigid evaluation.

Second, the recommender system that demonstrates the key combinatorial approach implemented in the system provides its functionality based on the user model, CBF and CF of user ratings. Since the user ratings for movies at the site are based on personal taste and preferences, it is impossible to quantify the value of recommendations made at the site. While the results of any content-based filtering can be verified by the simple analysis of whether the results contain the expected content, the value of results produced using collaborative filtering can only be reported by the user. The degree to which the recommendations are "good" recommendations is purely the user's opinion.

For these reasons, the qualitative testing effort hinges upon the development of valid testing procedures and the realization of appropriate criteria from which the results of these procedures can be evaluated.

5.1 Testing Procedures

As mentioned previously, system testing relies heavily on feedback from the user. These responses are collected from the user through a user survey, to which the user is asked to respond after testing the system. The qualitative testing process focuses on selecting the right user group, providing the appropriate instruction to the group and

presenting the proper questions on the user survey form to illicit meaningful responses for the project.

5.1.1 Test group selection and instructions

The group selected for testing the Movie Friend prototype consists of adult users, well educated in the use of computers, highly proficient with the use of multiple Web browsers and very familiar with common Web site navigation paradigms. Most of the users in the test group are Information Technology professionals. Still others are recruited from the various student computing facilities provided at Southwest Texas State University. These experienced users are selected out of a desire to minimize the need for familiarizing the user with the use of computers and Web browsers under the belief that these requirements are peripheral to the research project.

Most of the test group is recruited via a mass e-mail soliciting their input on the project. Other users are recruited in person, given verbal instructions, and asked to fill out the on-line Movie Friend Survey form. Those recruited by e-mail are given some initial instructions for the use of the site and then asked to complete the on-line Movie Friend survey (discussed in the next section) upon completion of the instructions. Users are encouraged to perform additional actions at the site if they desire before filling out the survey, and are encouraged to try to "break" the system by over-taxing it, entering erroneous data, attempting to manipulate other users' accounts, etc.

The instructions given to users consist of a series of simple tasks and are as follows:

- First go to the site at http://www.cs.swt.edu/~shane/moviefriend and familiarize yourself with the site (search for titles, list the top ten, etc.) first without creating an account.

- Create an account and WITHOUT RATING ANY MOVIES, list the movie recommendations for you by clicking on the "Your recommendations" link on the left of the page.
- Now RATE A NUMBER OF MOVIES and then check your recommendations again. As you rate more movies, you'll notice that you're affecting the ratings averages at the site and your recommendations should change over time.

Users are also asked to rate only movies they have seen and to rate them seriously, as ratings affect the ratings averages for the entire site and, consequently, recommendations to other users.

In response to the test group solicitations, 21 users create accounts at the site and perform various tasks. As evidenced by the tables of the database, an undetermined number of users access the site without creating accounts, and 15 users submit the requested Movie Friend survey form. It is these 15 users that provide the data discussed during discovery and analysis of the test results in the sections that follow.

5.1.2 The Movie Friend survey

The Movie Friend survey is implemented as an on-line HTML form that users are asked to fill out and submit after ending their test session at the site. It asks the user to enter his or her name and then poses the following questions with the indicated possible answers:

Q1: Did you enjoy using the Movie Friend site?

Yes, No, Don't know

Q2: Would you consider using sites like this in the future?

Yes, No, Don't know

Q3: Do you feel like your recommendations improved as you rated more movies?

Yes, No, Don't know

Q4: Does the adaptive nature of the site increase the likelihood that you will use it?

Yes, No, Don't know

Q5: If you were recommended movies that you have already seen but not rated, were the recommended films movies you liked?

Yes, No, No opinion

Q6: Would you trust recommendations from the site when selecting movies to watch?

Yes, No, Don't know

Q7: In what ways could the Movie Friend site be improved ?(Check all that apply)

More movies in the database; More features; Clearer purpose; Help for users; Access to other user's ratings/reviews; Improved interface

Q8: In my opinion, this project is: (Only one response)

Intriguing; Interesting; Valid; Dull; Pointless

Two additional text areas are presented on the survey from, allowing the user to suggest additional features and provide general suggestions about the site. Input from the user here is valuable and is considered but is not analyzed as part of the result analysis.

5.2 Evaluation Criteria

The criteria for evaluating the Movie Friend site relate directly to the questions posed in the Movie Friend survey described in the previous section. Prior to developing the survey, four specific metrics are identified as being key to determining in a qualitative way whether the Movie Friend Web site has achieved the goals set during project

development. These metrics – enjoyment, benefit, trust and interest – provide a clear framework within which to evaluate user responses. Even though the test group is relatively small, when considered in terms of the evaluation criteria, their responses lend a valuable perspective on the system implementation.

5.2.1 Enjoyment metric

One of the most fundamental goals, defined early in system development, is for the system to be enjoyable to use. While this metric might seem over-simplified, its contribution to any adaptive system seeking to retain users cannot be dismissed. Questions 1 and 2 of the Movie Friend survey clearly assess this particular metric directly, and provide insight as to whether the system is pleasing to our test group.

5.2.2 Benefit metric

The nature of the benefit metric, in relation to the Movie Friend prototype, is two-fold. Do users believe that there is a benefit to be gained from using the system to collaborate with other users, and do they see a benefit from the nature of adaptive sites? This benefit metric is assessed in questions 3 and 4 of the survey, and references the core of much of the functionality the system provides.

5.2.3 Trust metric

A level of trust must be attained for users to continue to use the site, this relates directly to questions 5 and 6. If a user believes that the recommendations at the site improve as more input from the user is accepted, the likelihood that they will continue to use the site increases. Likewise, intelligent users develop this level of trust, or distrust as the case may be, from all the data available to them. Obviously the system will periodically recommend movies to the user that the user has already seen but not rated.

This can contribute to the user's level of trust simply by the user's recognition that "I've already seen that movie and loved it." Conversely, it can lead to distrust by recommending a movie the user has already seen and disliked.

5.2.4 Interest Metric

The interest metric is primarily addressed in question 8, where the user describes his or her level of interest in the project using one of five keywords. Question 7 is used to gather valuable improvement suggestions from the user, but has the added benefit of indirectly providing insight on the interest metric. Presumably, disinterested users won't even bother to make improvement suggestions.

5.3 Responses and Analysis

When considered in terms of the four metrics discussed above, the 8 questions in the Movie Friend survey provide a significant amount of insight into the system prototype's implementation. Although simply user perceptions of the site, when explored in the context of the evaluation metrics, we get a reasonable idea of where the site has succeeded and where it has failed.

Furthermore, by examining the number and type of responses to each of the questions we gain a better understanding of what aspects of the project are implemented well and what needs to be improved. To aid this examination, the responses to each of the 8 questions in the Movie Friend Survey are reproduced in graph form below, and some additional observations are made.

Each of the graphs is titled with an abbreviated version of the question to assist analysis. Refer to section 5.1.2 of this work for the actual questions presented in the Movie Friend Survey.

The graph for question 1, shown in Figure 5-1 below shows the responses to the first of two questions that target the enjoyment metric.

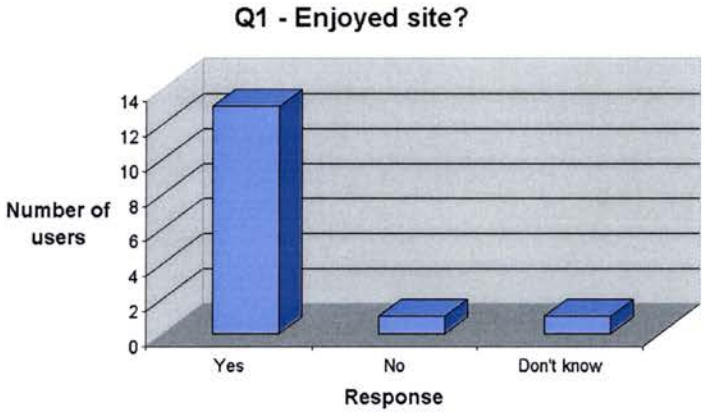


Figure 5-1: Survey question 1 responses

Figure 5-1 shows that 13 of the users in the test group said they enjoyed using the site, with one "No" and one "Don't know". While the test group is relatively small, this reflects a resoundingly positive response from the test group.

Likewise, in Figure 5-2 below, we see that only one user is uncertain about using the system in the future. Considered together in terms of the metric both of these questions assess, the responses to question 1 and question 2 provide a very strong affirmation that the system has met one of its primary goals, at least for the initial test group.

Compared with the results of evaluating the remaining metrics, this system enjoyability proves to be the strongest aspect of the system's design, as rated by the test user group.

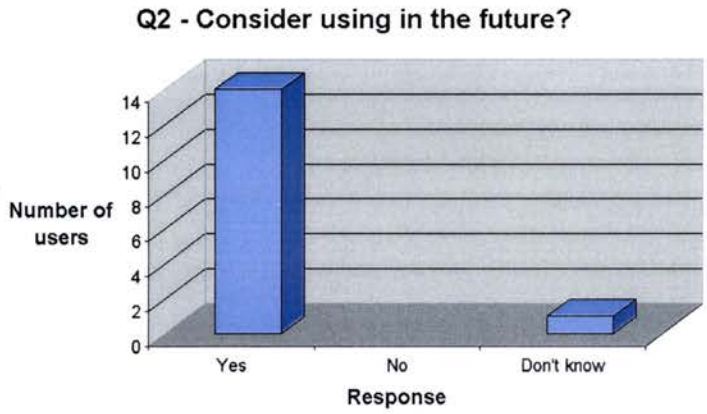


Figure 5-2: Survey question 2 responses

In Figure 5-3 below, the responses for the first question related to the benefit metric are shown. Here 9 users believed that their recommendations improved as they rated more movies, 2 felt that they didn't and 4 were unsure. It appears that users were somewhat skeptical of the system's recommendations capabilities. Indeed this skepticism is warranted in a system of this type with few users. As the system "matures", however, and more users and ratings are added to the system, it's reasonable to expect the recommendation features of the system will improve.

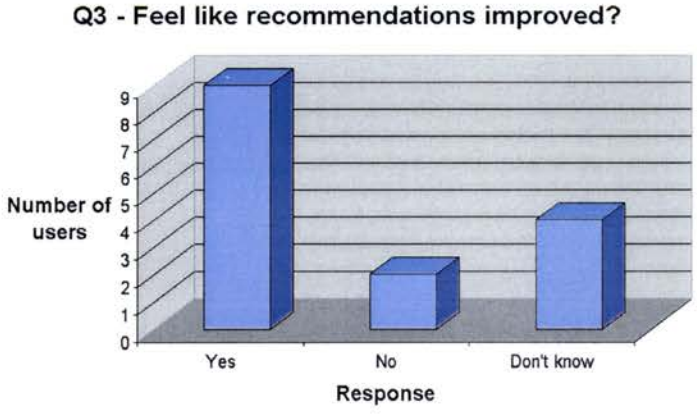


Figure 5-3: Survey question 3 responses

The responses to the second half of benefit metric assessment pair are overwhelmingly positive, as shown in Figure 5-4. All but one user agrees that the adaptive aspects of the site will increase their likelihood of using the site, one of the primary justifications made for developing adaptive Web sites in this work.

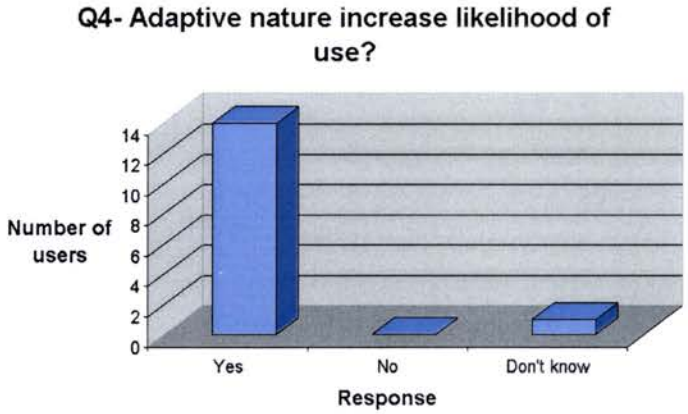


Figure 5-4: Survey question 4 responses

The user trust metric for the site is boosted by the responses to the first question used to assess it shown in Figure 5-5. Here 12 users indicated that when the system recommended movies they had already seen, they liked the recommended movies. Only 2 users said "No" and 1 was indifferent. This suggests that user trust is actually increased in cases where the system seems to the user to make an error. Perhaps the mere act of recommending movies that the user has already seen produces a more positive opinion of the system.

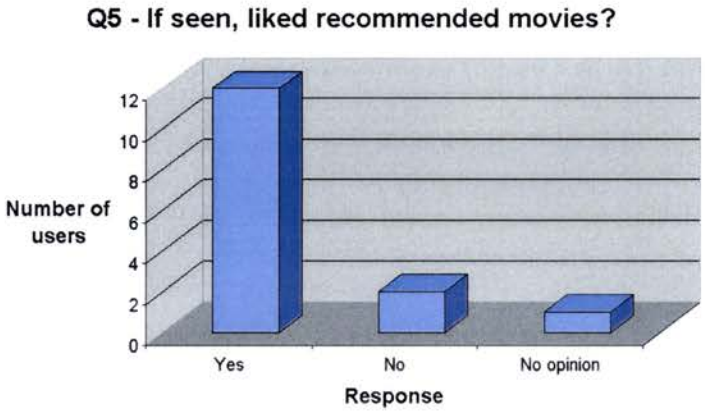


Figure 5-5: Survey question 5 responses

Unfortunately, the second question assessing the trust metric doesn't provide nearly as positive responses as the first, as shown in Figure 5-6. In this case, 8 users responded positively when asked whether they would trust the recommendations of the site when selecting movies to watch, 1 said "No" and 6 were uncertain. While not negative responses, these still seem to indicate that users are skeptical, primarily about the recommendation features of the site.

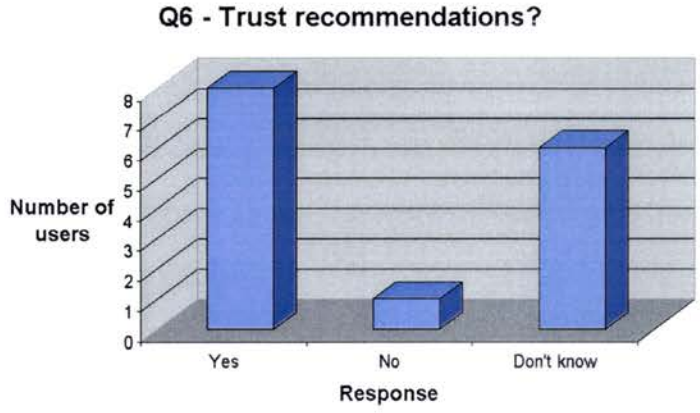


Figure 5-6: Survey question 6 responses

Our final metric, the interest metric is first assessed rather indirectly in question 7. Its responses are shown in the graph of Figure 5-7. It consists of improvement suggestions

for the site. Since users are allowed to respond to more than one category there are more than 15 total responses listed in the graph. From an interest perspective, what is apparent from this graph is that with 37 responses and 15 users, each user selected an average of 2.46 of the 6 options in the graph. If un-interested users don't provide improvement suggestions, this does imply a certain degree of interest.

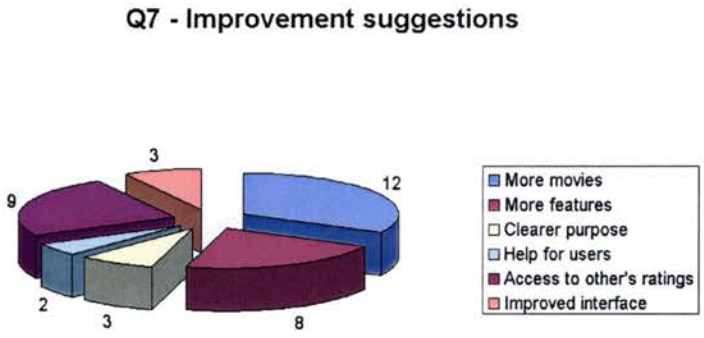


Figure 5-7: Survey question 7 responses

What is perhaps more telling for our interest metric are the responses to question 8, depicted in Figure 5-8 below. Here we see that 4 users rated the project "Intriguing" and 9 rated it "Interesting", the top two categories in the graph. Only 2 users rated it a middle-of-the-road "Valid" and none rated the project "Dull" or "Pointless".

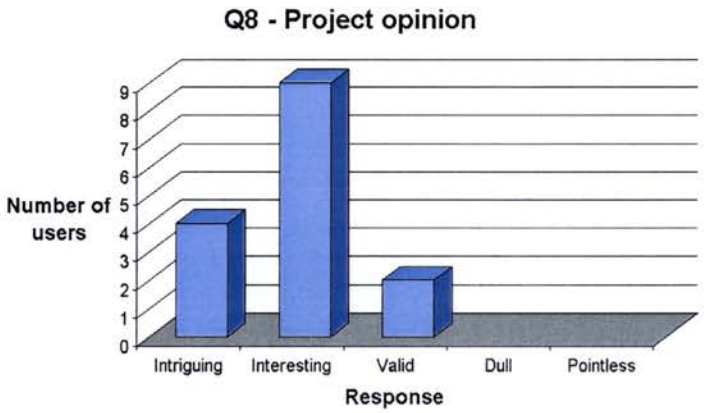


Figure 5-8: Survey question 8 responses

These graphs of the user survey questions, viewed with an understanding of the metrics used to generate them, provide a valuable view of the user test group's perspective of the system. Clearly, the project can benefit from their insight and strive to improve in the areas of user trust and interest, while maintaining the strength of enjoyment and benefit evident from the user test group.

CHAPTER 6

FUTURE RESEARCH

The potential for future research in the area of generating adaptive Web sites seems relatively wide-open, with a variety of curious special angles and "dark hallways" to venture into that this nor any other project researched as part of this one have explored. Indeed, as this project progressed it became a bit of my own curiosity to wonder about the potential of this project as a commercial endeavor. Not a for-profit endeavor, mind you. Rather, I could see the benefit of having such a system for entertainment purposes and could share it with the world - this being the true beauty of Web applications.

What this daydreaming did provide, if nothing else constructive, is a few thoughts about the potential of the whole area of what I'll call "adaptive systems." These can take the form of adaptive Web sites, as in the research of this project, or adaptive applications installed on a user's PC. Some areas of particular interest come to mind when considering the various challenges and theoretical potential that made this project intriguing throughout development.

In the area of adaptive Web sites, a system that uses all or most of the methods discussed in Chapter 2 of this work – one that uses user models, filtering techniques, page clustering, monitoring access patterns and data mining server logs would be an interesting undertaking. Granted, such a project would be huge in scope and developers

might soon find that some of the methods used in combination are detrimental to the end-goal. Even so, it would be a valuable study to discover whether these methods are contradictory or complimentary.

Research focusing exclusively on extending the recommender system portion of this type of project, in my mind would be a whole other, boundless, topic. There is any number of approaches that could be taken and tested just in the development of the recommender system "module" for an adaptive Web site. This research would no doubt include much within the field of Artificial Intelligence and the development of heuristics, etc. Presumably, though, one would need to build or have access to an existing fully-functional adaptive Web site in order to test the module, as much of the additional research in this area requires.

A third area for future research, specifically dealing with adaptive Web sites would be to explore methods for protecting the data at sites where collaboration is so heavily used. In sites, such as these, where users contribute much of the content to the site, it's necessary to develop methods to deal with users who seek to do harm, providing erroneous data and destroying data integrity.

Possibly the most intriguing thought that has come from this project is the idea of adaptive systems, or collaborative applications. It seems as though there might be a real potential for commercial software systems used to allow certain groups of people (say workgroups) to collaborate beyond the Web. In essence, it would be another form of productivity software customized for business workgroups.

This is not meant to suggest the development of custom software applications for businesses, as customized software is seldom considered cost-effective. Rather, the goal

here is to design "off-the-shelf" collaborative systems. These systems are then purchased and installed for business workgroups on a company network. Configuration might mean a process of entering user names within the workgroup, preferences and so on to customize the software, then users in the workgroup are able to collaborate with other users within the workgroup, similar profile, etc.

Admittedly, there are significant opportunities to improve upon the project implemented as part of this research and ample opportunities within the development of adaptive systems to find new and exciting areas to research.

CHAPTER 7

CONCLUSIONS

Adaptive Web site generation techniques differ greatly with differing results, but the imperative to provide users with dynamic content is undeniable for any Web venture hoping to survive in the vast network that is the World Wide Web. As evidenced by the statistics provided in this research project, users become habituated to sites that provide the best user experience, and provide the user with the greatest level of satisfaction when retrieving data.

This satisfaction relates directly to the key metrics – enjoyability, benefit, trust and interest - used in evaluating the Movie Friend prototype at the center of this research project. And our test group responses to the evaluation questions validate the prototype in terms of these metrics to varying degrees.

While the prototype is minimal and worthy of expansion, what is apparent is that the system works, and it works relatively well for the given test group. Granted, the system is still in its infancy, having a small number of users and a relatively small amount of data (in terms of movie records) with which to perform its adaptive functionality. Even so, if the lessons learned in other projects discussed in this work hold true, the system only stands to improve as more users join the system and as more movies and ratings are added to the system.

As mentioned in the previous section, an endless amount of work and effort could be devoted to improving the functioning of the recommender system component of the project, as many differing criteria could be imagined and implemented when collaborating similar users. This component, however, is really peripheral to the goals of the project, aimed at testing the feasibility and effectiveness of implementing an adaptive Web site system based on the combinatorial approach set forth.

In terms of the ability to generate adaptive pages based on content-based filtering, user collaboration and user models, this combinatorial approach is a definitive success. The extension of an explicitly defined user model in the system provides invaluable information to the system and clearly gives the system much more flexibility than is possible with an approach focusing on filtering methods and user ratings matrices alone.

Nevertheless, as with the recommender system, virtually endless work could be done to expand and empower the user model. It seems that many additional aspects of users could be added to the user model until a level of "diminishing returns" is reached.

Above all the research demonstrates that there is great potential through adaptive Web applications to increase user enjoyment, increase user value judgments about the data and services the site offers, and accordingly, to maintain those users. And the financial value to be gained by being one of the top "most popular" sites in a marketplace of billions of users is hard to overstate.

BIBLIOGRAPHY

[1] M. Balabanovic and Y. Shoham, Combining content-based and collaborative recommendation, Communications of the ACM, March 1997.

[2] M. Fernandez, D. Florescu, J. Kang, A. Levy, and D. Suciu, Catching the boat with Strudel: experience with a web-site management system. In Proc. of the ACM SIGMOD Conf. on Management of Data, pages 414-425, 1998.

[3] J.Fink, A. Kobsa, A. Nill, Towards a user-adapted information environment on the Web, in: Multimedia and Standardization 98: Paris, France, 1998.

[4] D. Fisher, Iterative optimization and simplification of hierarchical clusterings, J. Artificial Intelligence Res. 4 (1996).

[5] Global Reach. Global Internet Statistics. (n.d.). Retrieved April 8, 2003 from http://glreach.com/eng/ed/art/2004.ecommerce.php3

[6] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: Proc. AAAI-99, Orlando, FL, 1999.

[7] M. Kobayashi, K. Takeda, Information retrieval on the Web, in: ACM Computing Surveys (CSUR), Vol. 32 Issue 2, June 2000.

[8] A. Kobsa, J. Koenemann, W. Pohl, Personalized hypermedia presentation techniques for improving online customer relationships. The Knowledge Engineering Review 16(2): 111-155, 2001.

[9] A. Kohrs, B. Merialdo, Improving collaborative filtering with multimedia indexing techniques to create user-adapting Web sites, in: Proc. 7th ACM International Conference on Multimedia (Part 1) October 1999.

[10] A. Levy and D. Weld, Intelligent internet systems, Artificial Intelligence, vol. 118, no. 1--2, 2000.

[11] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering, In: Proc. of the SIGIR Workshop on Recommender Systems, 2001.

[12] B. Mobasher, R. Cooley, J. Srivastava, Automatic personalization based on Web usage mining, in: Communications of the ACM, Vol. 43, Issue 8, August 2000.

[13] NUA Internet Surveys. How Many Online?. (n.d.). Retrieved April 8, 2003 from http://www.nua.com/surveys/how_many_online/index.html

[14] M. Pastore. (2000, July 12). The Web: More than 2 billion pages strong. Retrieved April 8, 2003 from http://cyberatlas.internet.com/big_picture/traffic_patterns/article/0,,5931_413691,00.html

[15] M. Perkowitz, O. Etzioni, Adaptive sites: automatically learning from user access patterns, in: Proc. 6th Int. World Wide Web Conf., Santa Clara, California, April 1997.

[16] M. Perkowitz, O. Etzioni, Adaptive Web sites, in: Communications of the ACM, Vol. 43 Issue 8, August, 2000.

[17] G. Polcicova, P. Navrat, Combining content-based and collaborative filtering, in: ADBIS-DASFAA Symposium 118-127, 2000.

[18] U. Shardanand, P. Maes, Social information filtering: Algorithms for automating "word of mouth", in: Conference on Human Factors in Computing Systems—CHI-95, 1995.

[19] M. Tchong. (1998, July 1). Personalization statistics. Retrieved May 8, 2003 from http://www.iconocast.com/issue/1998070103.html

[20] A. Wexelbat, P. Maes, Footprints: History-rich Web browsing. In: Proc. Of the Conference on Computer-assisted Information Retrieval, pqges 75-84, 1997.

VITA

Michael Shane Flaherty was born in Ft. Stewart , Georgia, on April 8, 1969, the son of Julia Aileen Flaherty and Daniel Joe Flaherty. After completing his studies at Odessa High School, Odessa, Texas, in 1987, he attended Texas A&M University in College Station, Texas, Odessa College in Odessa, Texas, and Southwest Texas State University in San Marcos, Texas. He received the degree of Bachelor of Arts in Journalism from Southwest Texas State University in May, 1993. In June 1999, he returned to Southwest Texas State University and received the degree of Bachelor of Arts in Computer Science in May, 2000. In June 2000, he entered the Graduate School of Southwest Texas State University, San Marcos, Texas.

Permanent Address: 100 Camaro Way
San Marcos, Texas 78666

This thesis was typed by Michael Shane Flaherty.