Defining, Designing and Evaluating Social Navigation

Martin Svensson

A Dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy

Stockholm University

Department of Computer and
Systems Sciences

December 2002

DSV Department of Computer and Systems Sciences

Stockholm University
Royal Institute of Technology

Report series No. 03-001
ISSN 1101-8526
ISRN SU-KTH/DSV/R--03/1—SE
ISBN 91-7265-573-9

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Human Machine Interaction and Language
Engineering Laboratory, Kista

SICS Dissertation Series 33
ISSN 1101-1335
ISRN SICS-D--333--SE

Doctoral thesis
Department of Computer and Systems Sciences
Stockholm University

ISBN 91-7265-573-9
This thesis was typeset in Garamond 12pt using Microsoft Word 2000, Adobe Illustrator 9, and Adobe Photoshop 5.
Printed by Akademitryck AB, Edsbruk, Sweden, 2003.

Till Karin Ekberg

Abstract

The issue of how users can navigate their way through large information spaces is crucial to the ever expanding and interlinking of computer systems. Computer users live in a world of information spaces but in many situations lack the necessary means to navigate them. To meet this increasing need for navigational support *social navigation* has been proposed as a possible solution. The term captures every-day behavior used to find information, people, and places – namely by watching, following, and talking to people. This thesis sets out to investigate social navigation from three different perspectives: how it can be *defined*, how it can be *designed*, and how it can be *evaluated*.

By examining the properties of information spaces and navigation we define social navigation as navigation that is driven by the actions of others. Actions can by communication in several ways, and specifically, we make the distinction between direct and indirect social navigation.

Based on our understanding of social navigation five design principles for social navigation are proposed: presence, privacy, trust, integration, and appropriateness. They are issues that have to be considered when designing systems that support social navigation. The Social Navigator toolkit enables designers to in domain-relevant ways, instantiate and realize these design principles in their systems.

To test the social navigation design principles two socially enhanced food recommender systems were created. The first, EFOL, was evaluated in a small-scale study to verify that the Social Navigator worked. The second system, Kalas, was used to evaluate the benefits and problems with social navigation. In a study that ran over a six-month period Kalas was evaluated. The results partly indicate that social navigation adds quality to a system, that social navigation works well with other navigational aids, and that recommender systems need not be bootstrapped.

Acknowledgement

I want to mention the following people (in no particular order):

My advisors: Kristina and Calle.

For comments: Kristina, Rickard, Adrian, Åsa, and Jarmo.

For believing in the work: Kristina.

For support: Rickard, Kristina, and Anna.

For putting up with me: Anna, Agneta, Bosse, Hasse, Thor, Christian, Niklas, Alex, Johan, and Rickard.

For Kalas: Gerd, Anna S, Kristina, and Rickard

Contents

1 INTRODUCTION

This thesis is concerned with navigation in information spaces. Traditionally, support for navigation in computer based information spaces has focused on finding the shortest path between two locations, but there is more to navigation than this. Navigation can also be an experience: a user wants to travel the most beautiful or interesting path. It can also be a learning experience: a user wants to travel the most informative path. Navigation can also be part of the goal formulation process: a user may not have a clear destination in mind when starting out. Thus, design for navigation should not only be concerned with the shortest path, but with a whole series of other design goals.

Individual differences such as spatial ability and technical aptitude are some key factors that determine if a user is successful in navigating an information space. One way to deal with these differences is to incorporate various metaphors to structure the information in the space (Dieberger, 1994). For example, a library is known to most people and could therefore serve as a metaphor when structuring certain kinds of information spaces. These spatial metaphors make it easier for users to build mental maps over the information space, thus reducing the risk of getting lost. However, spatial metaphors have their limitations. No matter how well an information space is designed, some people will have problems finding their way around. These people use other means of navigation to find their way through an information space. We want to introduce social navigation as a metaphor for navigation. Our idea is to allow users to navigate computer based information spaces in much the same way as they often navigate the real world. Users should be able to use others to find their way around in various intricate and socially interwoven ways.

To do this we shall define what we mean by social navigation and discuss how it is possible to design for social navigation. We will develop a tool through which developers can add social navigation to their applications and show how the tool can be used to enhance a food recipe system with social functionality. Finally, we intend to show that social navigation does contribute to the users' navigational experience. We do this through evaluating the developed food recipe system using our own evaluation framework.

1.1 RESEARCH CHALLENGE

There is no longer a small group of expert users that use computers; most people use computers in their daily life. With the development of the Internet and the World Wide Web, the information that is accessible to people at any given point in time is enormous. The computer interfaces and programs of today are extremely complex and filled with information that people need to process. The problem is obvious: while software gets more complex to operate computer users are no longer specially trained operators but ordinary people.

Large multi-user information spaces will force the field of HCI (Human-Computer Interaction) to find new ways of designing systems. Much research has been devoted to solve the navigational problem users are faced with. Our approach is to introduce people as a fundamental tool for navigating a space.

The idea of social navigation is simple. Much of the information seeking in everyday life is performed through watching, following, and talking to other people. Why not create systems where we could watch, follow, and talk to other people, rather than navigate the space with maps and signs, to find what we are looking for? However, there is quite a difference between "natural occurring" social navigation in the real world and actually creating a design that allows for navigation in a constructed information space. Thus, we make two claims. First, that social navigation is a good design approach and that it is possible to map some of the social navigation we see in the real world to the electronic information spaces we are concerned with. The second claim, however, is that there is not a one-to-one mapping between navigation in the real world and navigation in the virtual world.

To understand how social navigation can be applied in computerized navigation, we need to solve three different problems. We have to investigate how navigation is conducted in the real world and transfer that knowledge into a working *definition* for social navigation in the virtual world. Second, we have to create a *design* model that we can apply to systems that supports social navigation. Finally, there has to be a framework for *evaluating* social navigation so that we can understand what works and what does not work.

1.1.1 DEFINING SOCIAL NAVIGATION

The concept of social navigation was introduced by Dourish and Chalmers in 1994. They saw social navigation as *navigation towards a cluster of people* or *navigation because other people have looked at something*. In parallel with their work, Hill and Holland invented the concept of edit wear and read wear (1992). By tagging information (or rather scrollbars) with read and edit patterns they affectively created the first history-enriched environments. Around this time collaborative filtering or recommender systems started to become popular (Shardanand and Maes, 1995). By collecting the opinions of a large number of people, an individual can specify one or two things that they like or dislike and the system recommends items based on the data collected from other people.

Later, Dieberger (1997) widened the scope set up by Dourish and Chalmers (1994). He also saw more direct recommendations of e.g. web sites and bookmark collections as a form of social navigation. He was inspired by the remarks made by Tom Erickson in 1996 that the web could be characterized as a *social hypertext* (Erickson, 1996), where nodes in it represent people. The links, as well as the page itself, provide us with a view of a person's network of friends, colleagues, and interests.

It is clear that social navigation can take many different forms, ranging from following a group of people that we do not know to approaching an expert in a field asking for advice on how to find information. One may distinguish between *direct* and *indirect* social navigation (Svensson, 1998; Dieberger, 2002). In *direct* social navigation, we talk directly to other users. In *indirect* social navigation, we can see the traces of where people have been in the space, as done in the Footprints system (Wexelblat, 1999). Social navigation could be *intended* or *unintended* by the users giving advice. An example of intended social navigation would be when we recommend to someone a place to visit, while paths through the woods can exemplify a situation where people do not intentionally leave traces for others to follow. Another distinction can be made between when the person giving advice is one particular person, known to us, or when it is just a group of anonymous people that have happened to navigate through the same space as us.

What is lacking is a definition that captures all the various forms of social navigation we see happening. What are the properties of social navigation and how are they related to social navigation in the real world?

1.1.2 Designing social navigation

Social navigation as a metaphor for navigation is complex and so far work on social navigation has mainly focused on understanding it, and not how to design for it. The only strong implication from previous work is that a system that supports social navigation should support *awareness* of people.

Underlying our design approach is the view that navigation should be a delightful experience, part of navigation is goal formulation, and we need to recognize the risk of making users anxious about getting lost or cognitively overloaded (Höök, 1998). In this, we break with the usability testing tradition that focuses on efficiency in terms of time spent and number of errors. Instead we focus on the quality of the experience. The visibility of users and their actions will become central. We will argue that there are five design principles that are of special importance when we want to promote the visibility of users in a system. These are:

Presence: how is the presence of other users mediated?

Trust: how can we trust the advice from other users?

Privacy: how can users' privacy be supported?

Appropriateness: when and how is social navigation suitable?

Integration: how do we integrate social navigation into an existing system?

1.1.3 Evaluating social navigation

Once we have an understanding of social navigation and how to design for it we need ways of evaluating whether it supports the navigational process. Many properties in social navigation are completely different from traditional user interfaces, rendering some of the existing HCI techniques for evaluation obsolete. In social navigation we are not only interested in efficiency (i.e. time to solve a task), we also want to get at issues such as pleasure, peoples' reactions to social navigation, and the cultural differences between people. Issues we are just beginning to explore and that are difficult to map to objective measures. Since the field is new, very few user studies exists that attempt to address these issues, but the following effects are discussed by us in (Dieberger et al., 2000):

Filtering: The purpose of history-enriched environments and recommender systems is to help users filter out the most relevant information from a large information space.

Quality: Sometimes it is not enough that the information obtained is relevant. It must also possess qualities that can only be determined from how other users react to the social texture. In many situations it will be other peoples' opinions that matter in deciding if something is useful or not.

Social affordance: Visible actions of other users can inform us what is appropriate behavior, what can or cannot be done. At the same time, this awareness of others and their actions makes us feel that the space is alive and might make it more inviting. Users can quickly pick up on the 'norms' for how to behave when they see the behavior of others.

Usage reshapes functionality and structure: Social navigation design may alter the organization of the space. It could be a first step towards empowering users to, in a natural subtle way, make the functionality and structure 'drift' and make our information spaces more 'fluid'.

We have to realize that even if the above mentioned properties are potentially very beneficial to users, there are also a whole range of potential problems with social navigation that might destroy the positive effects. The issues that we see as most important to address are:

Bootstrapping: Social navigation systems often rely on the accumulated user behavior, such as, trails of where people have gone. Such trails will work poorly when little information has been collected, and thus need to be bootstrapped before they can work properly.

Privacy: Social navigation relies on the visibility of people and their actions. In what circumstances are people willing to be visible and to what extent? There is reason to believe that some users are willing to share almost anything about themselves, while others want to be invisible.

Snowball effects: When more and more people walk down the 'wrong' path this will be indicated as a 'good' path in a typical social navigation system. To what extent is it possible to detect and deal with these wrong paths?

Concept drifts: Over time people and information change. In order for social navigation to be really successful it has to take into consideration that peoples' interests change and that different types of information have different expiration dates.

Design: Designing a social navigation system entails deciding on which of numerous ways social texture should be communicated, such as: how to mediate the presence of other users or finding the useful implicit actions that can be naturally included in the system dialogue.

1.2 METHODOLOGICAL OUTLINE

We have outlined the three research questions that the thesis will tackle. The research questions have to be addressed in different ways. To some extent they feed into each other. For example, when there is a definition of social navigation, it can be turned into a practical description of how to design for it in a particular domain. There are however certain methodological points we want to stress for each of the three research questions put forth.

Definition. We chose to adopt the view that social navigation is a way to support navigation. As such we needed to investigate what navigation really is, what the internal processes are when a person is navigating a space. This entailed some knowledge and definition of what constitutes a space or more specifically an information space. The research on navigation and information spaces is based on Downs and Stea's work on the function of cognitive maps (Downs and Stea, 1973).

Design. Designing for social navigation was grounded in our definition of social navigation. The design principles were turned into a set of basic functions that were subsequently made available in a designer's toolkit. Then, based on the toolkit and design principles, two systems for social navigation were developed, EFOL and Kalas. Thus, to verify that the design principles and toolkit are valid we used them to create systems that support social navigation.

Evaluation. The theoretical part of how to evaluate social navigation was verified against one small-scale qualitative user study and one larger, long-term study. The second study was a longitudinal real-world study. Non-parametric statistics (Siegel, 1988) were used in the larger study. Both studies were also used to validate the design principles.

1.3 CONTRIBUTIONS

Based on our understanding of navigation we propose a definition for social navigation, it is said to be *navigation that is driven by the actions of others*. We also make the distinction between *direct* and *indirect* social navigation, the difference between the two lies in the way actions are communicated. Several systems are also examined and classified within this basic framework.

In approaching the second objective we draw from our understanding of social navigation to propose a set of design principles for social navigation. We look at related work from other fields such as Computer Supported Cooperative Work (CSCW). Based on the principles we develop a toolkit (the Social Navigator) to form a framework that designers can use when they want to implement social navigation. Since we do not validate our framework through empirical studies we need to find another way of doing it. We approach the problem in two ways. We examine two other designers' toolkits for social navigation (WebPlaces and MetaWeb) and apply our design principles on them. Secondly, we use the Social Navigator and design principles to implement and design an online food store from scratch.

We will argue for a number of different ways for which social navigation are of benefit to a user. To validate these we conducted a real-world study – the Kalas evaluation – that ran for six months. The study is presented in such way that it can serve as inspiration for others studies of social navigation. The evaluation showed that social features are highly appreciated but not necessarily viewed as navigational aids by users. When asked, users often claim that it is the content that drives what information they choose, although it is clear that they use the social information to navigate a space. Furthermore, bootstrapping a recommender system is less of problem than it appears to be. A recommender system is used as a tool for inspiration and as such does not have to give perfect recommendations. Last, social navigation is not replacing but rather complementing other navigational aids. Social navigation is an additional way to find information and should not replace the other ways of accessing it.

1.4 PUBLICATIONS AND CO-OPERATION

This thesis is to a large extent the result of the i^3 project PERSONA (1998-2000) and SITI project PERSONAS (1999-2002). My role within each project has been to define and implement tools for social navigation. However, everyone involved has had some influences on the ideas presented in this thesis.

Chapter 2 on spaces, navigation and social navigation is my own work and the definition of social navigation is my own. The initial ideas on social navigation can be found in (Svensson, 1998).

Mattias Forsberg, Kristina Höök, and myself first formulated the design principles presented in Chapter 3 (Forsberg et al., 1998), but they have since then been heavily restructured. The other major source of inspiration in Chapter 3 was a meta-study performed by Kristina Höök and myself (Höök and Svensson, 1999) where the need for new measures when evaluating systems that support navigation was identified. The design and implementation of the Social Navigator is my own. However, Rickard Cöster and myself made the Social Navigator and recommender system (Cöster, 2002) APIs to fit the same basic framework, and as such, some of the architectural design decisions were a joint effort.

The online recipe shop (EFOL) presented in Chapter 4 is the work of Jarmo Laaksolahti, Kristina Höök, Annika Waern, and myself (Svensson et al., 2000; Svensson et al., 2001). There are, however, some points that should be made. Jarmo and Annika were the ones who developed the underlying recommender system. I did most of the work in implementing the other social navigation features in the store, especially in incorporating the Social Navigator toolkit and the design and development of the user interface. Finally, Kristina Höök did most of the work in the study of the online food store. For the Kalas system (Svensson and Höök, 2002) Anna Ståhl created the user interface while Rickard Cöster redesigned the whole recommender functionality.

Together with Kristina Höök the benefits and problems with social navigation were formulated. The ideas and discussion revolving how to evaluate them are mainly my own. Gerd Andersson, Kristina Höök, and myself set up the Kalas Evaluation. The evaluation analysis and results in Chapter 6 is mostly my own work.

1.4.1 READING INSTRUCTIONS

This thesis has three distinct goals and consequently has three parts. The first lays out the theoretical framework for social navigation, the second revolves around how to design social navigation, and the third, how to evaluate social navigation. The intention has been to make the parts as separate as possible so they can stand on their own. For the reader who is familiar with social navigation the first part is not crucial and for those who are not system designers the last part is of most interest. Part two and three are divided into a theoretical chapter and a practical chapter. The practical chapters apply the ideas on our chosen target domain: online-food shopping.

Part one. In Chapter 2 the theoretical framework on spaces, navigation, and social navigation is presented. The chapter starts with an examination of spaces and how they can be manipulated to serve different purposes. This is followed by a discussion on navigation. In defining navigation we try to answer three questions: "What do people know?"; "What do people need to know?"; and "How do people get their knowledge?", in order to navigate a space. Next social navigation is defined. We look at various ways of performing social navigation and the distinction between *indirect* and *direct* social navigation is made. The mode of communication between advice providers and seekers is defined. Collaborative filtering and history-enriched environments are also argued as ways of undertaking indirect social navigation. Several systems that implement social navigation are examined in detail. The chapter ends with a discussion how social navigation is related to other activities such as general socializing.

Part 2. Based on our notion of social navigation we propose a set of design principles and a toolkit for social navigation in Chapter 3. We will argue that the design principles need to be considered when designing for social navigation. The Social Navigator toolkit is designed and implemented. This part serves two purposes. It is intended as a way to verify that the design principles can be implemented and it is a tutorial on how a social navigation toolkit can be used. This part is practical and can be skipped by those who are not interested in what type of functionality a toolkit has to have to be a useful tool. The chapter ends with a comparison between the Social Navigator and two similar systems: MetaWeb and WebPlaces.

In Chapter 4 the online food shopping domain is examined. EFOL, a recipe recommender system, is designed based on the design principles and the Social Navigator. The results of a small study of EFOL are then used to redesign the system, resulting in the Kalas system. Chapter 4 can be viewed as the practical application of the theoretical framework developed in Chapter 3. The small-scale study presented in Chapter 4 is targeted towards showing that EFOL supports social navigation (or is designed for social navigation) rather than evaluating if social navigation actually is of benefit to a user.

Part three. Chapter 5 are concerned with how to evaluate social navigation. In short, Chapter 5 puts forth several dimensions in which we see social navigation as a potential benefit to people. In the chapter we argue why social navigation could be used as a tool to filter information and why it can support social affordance, to name two such dimensions. On the basis of our understanding of the potential benefits and problems with social navigation we show several ways of evaluating them. The purpose of the chapter is to keep the discussion general, making it possible to apply it to various practical situations.

Chapter 6 applies the ideas in Chapter 5 to a real-world scenario. It presents a larger study of Kalas. While the smaller study presented in Chapter 4 focused on how to design social navigation, the Kalas study is targeted towards how to evaluate social navigation. Based on the evaluation criteria developed in Chapter 5 we ask the question: “Does social navigation work in the online food recipe domain?”. Chapter 6 also goes to some length to provide the reader with inspiration for how to evaluate social navigation.

2 DEFINING SOCIAL NAVIGATION

How can we empower people to find, choose between, and make use of the multitude of computer, net-based and embedded services that surround us? How can we turn human-computer interaction into a more social experience? How can we design for dynamic changes in system functionality based on how systems are used? We observe that much of the information seeking in everyday life is performed through watching, following, and talking to people. When navigating cities people tend to ask other people for advice rather than study maps (Streeter, 1985), when trying to find information about pharmaceuticals medical doctors tend to ask other doctors for advice (Timpka and Hallberg, 1996). Munro observed how people followed crowds or simply sat around at a venue when deciding which shows and street events to attend at the Edinburgh Arts Festival (Munro, 1999).

There is only so much information we are able to process and filter out. At some point spaces become too complex and we loose our bearings, we become lost. The problem becomes crucial in the computer based information spaces we typically find in our everyday life – word processors, spreadsheets, or the World Wide Web. They are so complex and information-dense that many people find it impossible to locate the information they need. In a space such as the World Wide Web, a user can spend hours trying to find the information she is interested in. We can view this activity of finding and filtering the information in these spaces as navigation. What we do is to navigate among the bits of information to find what we want.

In order to understand social navigation it is essential to investigate the activity it supports – namely navigation. Also, it is necessary to pin down the notion of an information space and what we mean when we talk about an information space.

2.1 INFORMATION SPACES

An information space is anything that allows information to be stored, received, and possibly transformed (Benyon and Höök, 1997), thus more or less anything can be regarded as such. Information spaces come in many flavors, they can be physical (e.g. airports or conference centers), or they can be computer based such as word processors. A typical view of a space is that it is something that we can physically *move* around in. We place no such limits on an information space: a movie is regarded as an information space since it is a bearer of information – although fictitious. In the same way we move around – although not physically - in a word processor by clicking a mouse.

We move between different spaces and we find spaces within spaces. The office constitutes one space with a set of physical boundaries, while the World Wide Web (hereafter named the web) is another space with different boundaries. Physical information spaces are always spatial Euclidean spaces, since the world is by definition Euclidean. A virtual space, on the other hand, can be both: a VR environment is spatial (and may sometimes obey Euclidean laws in a "virtual" sense if implemented to do so) but a word document would not be regarded as a spatial space in the Euclidean sense, although we may perceive them both as spatial.

The real world (i.e. the physical world) consists of a number of spaces, hence it makes sense to look at the properties of those spaces when defining what constitutes a spatial space. Based on the real world, Harrison and Dourish (1996) list a number of properties that have to be satisfied in order to classify a space as spatial:

- **Relational orientation and reciprocity**. The world is organized in the same way for all of us, at least at a cultural level. Different cultures view the world differently, but within a culture we have the same frame of reference (Lynch, 1973). Down, up, center, have the same meaning for everyone, or as Harrison and Dourish (1996) put it: "Since we know that the world is physically structured for others in the same way as it is for ourselves, we can use this understanding to orient our own behavior for other people's use".
- **Proximity and action**. We act in our near proximity. People have physical limitations, it is not possible to pick up an object far away or carry a car. Our understanding of the world makes it possible to infer that a car that looks small is further away than a car that looks big.
- **Partitioning**. Interaction occurs in our near proximity so distance can be used as a means to partition a space and interaction. Partitioning can be made stronger with psychical boundaries (e.g. a house is divided into rooms).
- **Presence and awareness**. The real world is filled with more than artifacts, it is inhabited by people and their traces as well. The awareness of others and their actions influence our own activity.

Harrison and Dourish studied collaborative systems where people are naturally part of the spatial space. Although people are central to our work we still regard spaces without people as spatial ones, in effect, the last bullet is important but not central to such spaces.

It is clear that physical information spaces are spatial spaces and in general VR-environments or CVEs (Collaborative Virtual Environments) adhere to the properties that constitute a spatial space, thus we classify them as spatial. A spatial space is often visual and 3D based, although this is not necessary. It is possible to create a text-based MUD[1] environment that is spatial as long as the textual description and properties of it conforms to the properties of a spatial space.

[1] Multi User Dungeon. A role playing game that takes place in a real world setting. MUD's are usually multi user and text based.

In comparison, non-spatial spaces are everything else. Typically such spaces are found in the virtual world. Both a MUD system and the web would be regarded as non-spatial, although they seem to be totally different, one utilizing spatial characteristics and the other more semantic ones. There are a number of properties that can be modeled to change the appearance and usage of a space, and this is the reason why the two examples above are different. According to the definition, a typical MUD is not a spatial space, but uses spatial metaphors from the real world to make it seem like one.

2.1.1 Environmental knowledge

People acquire knowledge about large-scale spaces in three steps (Siegel and White, 1975). First, people recognize landmarks in the space, i.e. they acquire *landmark knowledge*. Landmarks are distinctive features in the environment that people use as reference points. Second, landmarks are connected into routes. This is refereed to as *route knowledge*. Routes are procedural description on how to get from one place to another. People are able to navigate a space on given routes but would be lost if they move outside the boundaries of the routes. Finally, people acquire *survey knowledge* of a space. This is often referred to as having a *mental map* of the space or a "birds eye" representation of it. With survey knowledge of an environment people are able find alternative routes and shortcuts. People have accurate knowledge of where objects are, distances between objects, and how they are related to each other.

2.1.2 Properties of space

There are more dimensions to a space than just a spatial one. A space can have a semantic structure or a social structure (Dourish and Chalmers, 1994). Spaces can be dynamic or static. A single-user space is different from a multi-user space. In this section a set of properties are listed that can be manipulated in order to change the appearance of a space. That is, there are a number of ways in which a space can be structured so its inhabitants will perceive it differently.

One has to realize that computer based information spaces – which is the focal point of this thesis – are just electronic bits of information that are put together to serve some purpose. In the same way as a child has to learn the properties of the real world, users have a varied understanding of virtual information spaces, and thus, perceive them differently. The spatial metaphor that a virtual space is modeled around may not be apparent at first. The same holds for the semantic relations in a hypertext system. In other words, constructing a virtual information space with a metaphor borrowed from the real world (e.g. a house) does not necessarily mean that users will perceive it as such a space.

Metaphors. Although a space is not spatial by definition, it is possible to use spatial metaphors when designing it. The hypothesis is that metaphors from the real world make the transition to the virtual information space easier for people. Structuring, for instance, the computer interface as a desk makes it possible to structure information in the space around folders, documents, and trashcans. Then, the argument is that users will immediately understand that documents can be placed in folders or thrown in the trashcan, from their knowledge about a real office.

An even stronger metaphor is Dieberger's (1994) virtual city. It is an attempt to organize information around objects and concepts found in an ordinary city, allowing a user to navigate and interact with the information space in the same way she would do in the a real city. In the virtual city it is natural to store private information in houses. Places where people interact could be represented as parks, and streets could be used to move between objects. The information space is structured in a way that comes natural to people, allowing them to take advantage of their knowledge about a city.

Gentner and Nielsen (1996) argue that it can be problematic to design and structure spaces around metaphors. The purpose with metaphors is to use concepts already know to users (as the desktop metaphor mentioned above), thus releasing some of the burden from the user when she acquires knowledge of a space. However, as Gentner and Nielsen point out, using known metaphors can limit the design of a virtual space. Specifically, there are three issues that need special attention (Halaz and Moran, 1982):

- Features found in the target domain do not exist in the source domain. In MUDs users can use instant transportation, a feature not found in the real world. If the connection between the real world and the MUD is made too strong, it could be the case that users do not look for the instant transportation feature found in MUDs.
- Features found in the source domain do not exist in the target domain. In the real world it makes sense to talk about Euclidean distance, but in MUDs it is not applicable to measure the Euclidean distance between two locations.
- The source and target have features or functions in common that work very differently. Both in a MUD and in the real world people can go left, right, forward, and backward. However, in a MUD it is not always the case that if a person goes left and then right she ends up where she started.

Since we live and act in spatial space we have a strong tendency to use metaphors when we talk about spaces, independently of the space we are talking about. Maglio and Matlock (2002) show that web users remember and talk about the web in the same way as they do with physical spaces, e.g. they describe the web in terms of landmarks and routes.

Semantically organized. A space often has a semantic structure; objects in the space are related to each other. Grouping related objects makes it easier for users to find their way in space and it makes the space seem more organized. In the same way that there is a common understanding of spatial relations there is often an agreed upon understanding of semantic relations. To organize a space semantically does not only imply grouping related objects, but also to include the proper objects in the space. Of course, semantic organization is something that to a large extent is learnt. The fact that a user knows that the 'edit menu' in a word processor should contain the commands 'copy', 'past', 'edit', and 'search' is not so much about their semantic relationship per se, but more an outcome of a de facto standard on how to build a word processor.

In non-spatial spaces where many of the ordinary navigational tools and skills are of less use it is important that the space is organized in some other way. No matter if the space is virtual, physical, spatial, or non-spatial, people need a common understanding of it. Since a semantic relationship does not rely on any underlying spatial organization it makes sense to design a non-spatial space with strong semantic relations between objects in the space.

Hypertext systems (Conklin, 1987) utilize the idea of semantic relations. The space is built up by bits of information that are linked to each other by ways of semantic relationships. Instead of navigating in terms of ordinary Euclidean distances and compass directions, users move between objects in terms of their semantic relations.

Socially organized. It is not the semantic and spatial properties of a space that tell people how to act in space. We can think of it as the spatial and semantic properties mark the boundaries of the space to the rest of the world, but something else informs us how to act in it. To cite Harrison and Dourish (1996) "Space is the opportunity; place is the understood reality". What they mean is that *space* marks the physical boundaries and it is the *place* that brings meaning to the space.

The same underlying space can yield several different places. At daytime the conference hall is a place where people meet and discuss research and during nighttime it is turned into a reception area where people socialize and have fun. Thus, the same space can over time be differently organized in terms of social behavior.

Social connotations are what constitute a place (Dieberger, 2002). They can change over time and are weaker than the physical boundaries of a space. In the real world it is often possible to tell from the outside what social connotations constitute a place. By observing a pub we can tell how to behave. In the same way, we draw the conclusion to not shout in a library since everyone in it is quiet. On the other hand, social connotations are subtler in virtual information spaces. Typically, in virtual information spaces, we cannot see people and how they interact until we join them. As Dieberger (2002) notes one major challenge for virtual places is to make them more visible from the outside.

Dynamic versus static. Lastly, spaces can either be open (dynamic) or closed (static). Closed spaces are stable in time, i.e. they seldom expand or collapse. Conversely, an open space changes over time and users in it constantly have to rebuild their mental representation of it. Open spaces are harder to navigate than closed spaces and the tools that can be used to support navigation in those spaces are fewer – maps turn obsolete, landmarks are removed – than in the closed spaces. Suppose that we want to use an agent that explains to a user how an information space is organized. In a dynamic space the agent can quickly turn obsolete and instead of helping the user navigate the information space, the agent only hinders her. Search engines on the web have this problem, they often give users information about nodes that no longer exist. This is due to the web's dynamic structure – nodes are constantly added and removed.

2.2 NAVIGATION

One of the fundamental activities that we conduct in space is navigation. There are other actives as well, but without the ability to find the way in space we would be inherently lost and thereby unable to perform other tasks. We navigate both when we are in a new space but also in our everyday life (without thinking about it consciously). Going to work each morning has some element of navigation to it, especially if the ordinary route for some reason is not feasible and an alternative route has to be chosen.

The reason for getting a better understanding of navigation is to, in the end, aid users when they navigate a space. First, navigation takes time. People spend a lot of time searching for information and if we could reduce the time it would be a gain to users. Second, we want to reduce the feeling of being lost. Anyone who has been lost in an unfamiliar environment knows that the feeling is unpleasant and irritating. Third, navigation is a process. There is more to navigation than just moving from one place to another. When we navigate we may reformulate our destination, we experience the space, and we evaluate our destination with qualitative measures. Fourth, navigation should be fun. Not only should we reduce the feeling of being lost but also make navigation a pleasurable experience. Finally, navigation is learning. Time is not the only important factor, sometimes we navigate a space to understand it and in such situation it is not always the case that we want to find the shortest route but rather the most informative route.

In short we define navigation as *the activity of going from one place to another*. As a basis for the following discussion we will use Downs and Stea's work on cognitive mapping (Downs and Stea, 1973). Cognitive mapping is defined as:

> *"Cognitive mapping is a process composed of a series of psychological transformations by which an individual acquires, codes, stores, recalls, and decodes information about the relative locations and attributes of phenomena in his everyday spatial environment"*
>
> Downs and Stea, 1973

In their discussion on cognitive maps Downs and Stea list three questions that need to be answered in order to understand the function of cognitive maps: (1) What do people need to know? (2) What do people know? (3) How do people acquire their knowledge? Although our purpose is not to investigate the nature and function of cognitive maps and Downs and Stea did not explicitly discuss navigation, there are similarities between navigation and the purpose of cognitive maps. It is apparent that a cognitive map supports a user in finding her way in space.

WHAT DO PEOPLE NEED TO KNOW?

Given that a user wants to navigate from one place to another there are two fundamental bits of information she needs to posses. First of all, a user has to know where she wants to go, that is, she must know the *location* of her final destination. Locational knowledge allows a user to orient herself in the space that she is in (what Downs and Stea refer to as *keying*), which is extremely important in navigation. If a person cannot tell where she is, how could she possibly formulate a route to her destination? Obviously, it is not enough to know the starting point and end point. A user should be able to key herself when moving towards the final destination, i.e. in navigation we also have to *monitor* a given route.

Secondly, a user has to be able to identify the destination. The salient *attributes* or *characteristics* of the destination have to be clear to her. Attributive information tells users something about the destination that they want to reach. As an example we can imagine a user that wants to go from her hotel room to the main conference hall. She uses her map of the conference center and ends up in a small room that is empty. From the attributive information she has of the hall (it should be large and full of people) she concludes that she is lost. Downs and Stea identify two types of attributes:

- *Denotative* or objective descriptions of an object. The attributes that describe an object in terms of what it contains or what it looks like. For instance, an airport contains planes, counters, and baggage claims. One could argue that there are no such things as *objective* attributes since all objects are perceived individually, subjectively recognized, and also, depending on the culture the same objects can have different descriptive attributes. However, at some level of abstraction we have a common understanding of what constitutes an airport.
- *Connotative* or evaluative attributes. Evaluative attributes are more subjective and cannot be read out directly from objects. They are more dynamic than denotative attributes and change over time. Again take the airport as an example, typical connotative attributes could be "it sticks to schedule" or "it has a very good tax free shop". It is not always enough to know what an object looks like or where it is located. It is, for instance, difficult or impossible to find the 'best' airport using a map as the only navigational aid.

An airplane is a denotative attribute of an airport, but is not an airplane an object of its own with distinctive attributes? Depending on the level of granularity, what is an object in one situation might be an attribute in another. Put in another way, the space a user is currently navigating and the destination she wants to reach defines what constitutes an object or attribute. Let us use the conference example to clarify. If a person wants to find the conference center (the object) the "main" conference room is a descriptive attribute of it. On the other hand, if she wants to find the main conference room, the room is the object and chairs would be descriptive attributes.

WHAT DO PEOPLE KNOW?

What knowledge do we have at our disposal when navigating a space and what might that knowledge look like? Our representation of a space (or the world we live in) is incomplete. The fact that a person does not know what is behind the horizon does not imply that the world ends at the horizon. Furthermore, a person's view of a space is often *distorted* in terms of distance. For example, the distance to work may seem longer than going home from work. We will also use (conventional) symbols as a means to talk about the world. The Great Wall of China can be used in our representation of the world, even though we have never been there or will go there. Related to this is the fact that we generalize our knowledge: "If you have seen one conference center, you have seen them all".

Finally, there are both *individual* and *group* differences in the way we perceive the world. Two different cultures can have different ways of drawing maps of the world. It does not necessarily mean that they have a different understanding of the world, but the knowledge of the world cannot be communicated since they use different ways of describing it. Similarly, the statement "it is a short walk" can have a different meaning for two people.

HOW DO PEOPLE ACQUIRE THEIR KNOWLEDGE?

When navigating the real world a person can use all her senses to find the way. People know that they are near the airport since they can hear airplanes taking off. The distinctive "sea breeze" is an indicator of being close to the sea. In the virtual world a user is often limited to visual and auditory input. Additional to this very fundamental way of learning and navigating a space a person has other means to gain knowledge, what Downs and Stea refer to as vicarious (or second hand) sources of information. These include maps, talking to people, reading signs, and so on. The distinctive feature of this type of information is that someone else other than the person using it has filtered it in one way or the other. In a way, the sensory modalities let us perceive a space as it really is and not through someone else's eyes.

To summarize the discussion we extend our proposed definition of navigation. It is now clear that there is more to navigation than just moving from one place to another. When looking at what people need to know in order to navigate a space, it becomes apparent that it is not enough to know the start and final destination. Based on Satalich (1995) we extend our definition of navigation and take it to mean:

- Orienting oneself in the environment
- Choosing the correct route
- Monitoring the route
- Recognizing that the destination has been reached
- Choosing or formulating a destination

Consequently, there are five activities that make up the navigational process and these are the activities we have to support. It is not enough to aid the user in choosing a correct route. It is equally important to aid the user in monitoring a given route, aiding her in recognizing that the destination has been reached, and to assist her in orienting herself in the environment. It is not always the case that we have a clear picture of where we want to go. On the contrary, we often only have vague ideas on what it is we want to accomplish and where it is we want to go.

2.2.1 TYPE OF NAVIGATION

We adopt Benyon and Höök's (1997) notion and divide navigation into three different but kindred activities: *wayfinding*, *exploration*, and *object identification*. They are really different sides of the same coin – it is only the purpose of the activity that differs somewhat.

Wayfinding

Wayfinding can be characterized as the activity of going from one place to another. A user has a certain destination that she wants to reach. When we think about navigation we normally think of it in terms of wayfinding. In wayfinding the activities outlined above are equally important. A navigator has to orient herself in the environment, choose a route, monitor the route, and finally, recognize that the destination has been reached.

Exploration

Intuitively there is a difference between wandering around in the conference exhibition area compared to actively search for a specific booth. Navigation without a specific destination is called exploration. In exploration people are not so much interested in a specific location, but more interested in exploring the space they are in. What are the interesting objects? In exploration people are more open to following a crowd of people or randomly choosing a route.

In exploration the destination and correct route are of less importance. To be able to orient oneself in the space is still important, else people would be lost, which is not the same thing as exploring a space.

Object identification

A space consists of a number of objects with specific attributes that serve to identify them. To perform successful wayfinding people have to be able to identify objects in space or at least the object (location) they want to reach. To find a conference center a person has to know what a conference center looks like.

When people are in a new space the first thing they have to do is to identify the different objects that a space consists of and their respective attributes such as identifying the landmarks. In a way this activity is no different from wayfinding, the only difference is the space that is navigated. To successfully identify an airport we first have to navigate within that space to find the connotative features and how they are related.

It is not the case that a person always engages in object identification when she enters a new space. People have the ability to generalize from past experiences, thus, if a person knows what a conference hall looks like at a specific conference center she can apply that knowledge when she wants to find the a conference hall in another center. Denotative attributes tend to be the same for the same type of objects in different spaces. An airport in Sweden and an airport in England both have airplanes, check-in counters, and baggage claims.

It is harder to generalize from connotative attributes. Connotative attributes are not built into objects in the same way as denotative ones. The people who use the objects create them. As such the attributes will vary over time. The statement "Once you have seen one good pub, you have seen them all" makes little sense.

2.2.2 Navigational aids

What Downs and Stea refer to as vicarious sources of information for acquiring knowledge about a space we call navigational aids. A user seldom has complete knowledge of a space, and hence, she has to use various navigational aids (or external sources of information) to find her way in space. Typically these are maps, landmarks, and signs.

Landmarks. Landmarks are objects in a space that serve as reference points to people (Lynch, 1960). They are salient features of the environment. Landmarks can either be personal or shared. The Statue of Liberty would be an example of a landmark that is shared. Personal landmarks are objects that have a special meaning to some specific individual. Landmarks are the basic building blocks of our mental representations of a space. Since we seldom have complete knowledge of a space we use landmarks as a means to structure space. They are key points that we can turn to when we are lost, or when we have to re-orient ourselves in the environment. Landmarks are often used in other navigational aids, for example, they are often highlighted in maps and people often use them when they give navigational advice.

Signs. McCall and Benyon (2002) identify three types of signs: *directional*, *informational*, and *warning*. Directional signs provide route or survey information. These types of signs are intended to guide a user in certain direction. Informational signs are used to describe objects in a space, easing the burden for a user when she is engaged in object identification. The warning signs provide information on the potential actions in the space. Signs are often used as a navigational aid in complex environments. Airports frequently use signs as a means to aid users in finding their way in the environment. In one study signs were found to be an effective way to guide people in the right direction (Butler et al., 1993).

Maps. Maps are used as external representations of space. When people think of navigational aids, they probably think of maps. Maps are used in all sorts of navigational situations, ranging from seafaring to aiding newcomers in large buildings. Maps are survey representations of an environment and we typically find two variants of the basic map: *you-are-here* maps (a map that marks the position of the person looking at it), and *route* maps (maps with an explicit route marked). The major problem with maps is that they are complex. It usually requires a lot of training to be a good map-reader. Butler and colleagues (1993) found signs to be more effective than you-are-here maps. Vocal directions are also a more effective navigational aid than route maps (Streeter et al., 1985).

2.3 Social navigation

As we saw in the previous sections navigation is a fundamental part of our everyday life. For obvious reasons the concept of navigation is not new. People have navigated the world for ages, and the techniques and skills for finding the way have become more and more sophisticated. The art of map making is more accurate then ever before, and the development of the GPS (Global Positioning System) allows us to pinpoint our exact location wherever we are. However, people typically do not carry maps with them (apart from when they are in a foreign city) and GPS receivers are rarely used in everyday life. How then, do people navigate? We have argued that people do not have complete knowledge of the spaces they are in, so navigational aids are indeed necessary to find the way. Obviously, there are landmarks and signs built into the spaces we navigate, but are they enough?

When looking at the way people navigate the real world, it becomes apparent that they are themselves crucial navigational aids. That is to say, we often use other people when navigating. Let us illustrate the point we are trying to make:

> ***The Baggage Claim****: Mr. Smith lives in a small town in Scotland and is going to London for the weekend. Since this is his first trip by airplane he is a little nervous. Anyway, when he arrives in London he is unsure of where to find his luggage. However, since he's been around he decides to follow the crowd from the plane (they ought to have luggage as well). Five minutes later Mr. Smith stands at the baggage claim to pick up his luggage.*
>
> ***Finding the Pub****: Mr. Smith is really enjoying his first holiday in London. The city is huge and there are lots of things to look at. He is a little disappointed though, he hasn't found any good pubs (the ones suggested in the guide feel like poor imitations). Suddenly he realizes that his old friend from back home lives in London. Mr. Smith calls him up and asks if he knows of any good pubs. "What sort of pub?", the friend asks. "A good one", Mr. Smith replies. "Ok, like a Scottish or English or maybe an Irish pub?", the friend asks. "Ah, a Scottish one", Mr. Smith answers. The friend suggests the Old Scotsman, a pub that should be just in his taste. The pub turns out to be just what Mr. Smith was looking for.*

Several interesting phenomena are disclosed in the examples just given. First it is noted that the advice is communicated in rather different ways in the first and second story. In the first story Mr. Smith follows a crowd of people and he is not explicitly asking for advice. He also draws his conclusions from the fact that there are several people going in the same direction. On the other hand, when Mr. Smith wanted to find the best pub he directly contacted his friend for advice. His friend also asked him to clarify his goal (i.e. Mr. Smith wanted to go to a Scottish pub). The two stories are examples of a family of navigational strategies that is called social navigation.

2.3.1 DEFINING SOCIAL NAVIGATION

The difficulty in providing a clear-cut definition lies in both the navigational part of the activity and the social part. Computer based information navigation is not easy to understand. What does it mean to navigate a word processor? While we might come to an agreement on what navigation entails as well as how we understand an information space, adding the concept social complicates things. What does social mean in this context? It could indicate the use of man made navigational aids, or entail some form of direct communication between two or more people. Can Agents act social towards humans? Another issue that needs to be resolved is whether we are more interested in supporting social interacting than aiding navigation.

If we define social navigation to be navigation with man made navigational aids, virtually any type of navigation would have to be regarded as social; for example, navigating with help of a map would be social. On the other hand, it is not enough to take it to only mean interaction between people to solve a navigational task. To follow someone else's trail is something we regard as a typical example of social navigation.

Although difficult to define, it is necessary to have a common framework when talking about social navigation. Merely the fact that there is no agreed upon understanding of social navigation makes it vital to understand what we mean by "navigating socially". It is therefore timely to offer a definition of social navigation that will be used as a starting point for the following sections, but before doing so we note that there are two parties involved in social navigation. The *navigator* is

> *the person seeking navigational advice. The navigator is also referred to as the user or advice seeker.*

An *advice provider* is the

> *person or artificial agent providing navigational advice to a navigator.*

The important thing to observe is that an advice provider can be an artificial entity, i.e. social navigation does not necessarily have to be based on people. As long as the navigator sees the artificial agent as another habitant of the space, the agent can serve as an advice provider. Consider the following two examples: (1) a navigator is following the trails of a MUD robot; (2) a navigator is following another MUD player. The difference is that in the first case a user chooses to follow an agent and in the second a real player, but would we regard the style of navigation any different? The fact that the navigator chooses to base her navigation on the *actions* of another (be it an agent or human) would in this case be enough to call it social navigation. We define social navigation as:

> *navigation that is conceptually understood as driven by the actions from one or more advice providers.*

When a user chooses to base her navigation on what others have done or the advice provided by others, it is social navigation. Social navigation does not have to be based on an advice provider per se, as long as the user believes she is following the actions from an advice provider. This means that a user who chooses to borrow a book from the library based on the fact that it looks well read is navigating socially, even if the book in fact has been tampered with to look worn. What is conceptually understood as actions of others is, thus, included in our definition of social navigation.

MODE OF COMMUNICATION

There is not only one type of action that can be used in social navigation. Actions can be direct advice from an advice provider to a user, following a crowd of people, or aggregated usage (such as paths in the woods). A fundamental difference between ways of undertaking social navigation is the mode of communication between a user and advice provider, i.e. how the advice provider's actions are communicated to a user. We therefore define direct social navigation as social navigation where

> *communication between navigator and advice provider is mutual and two-way.*

In indirect social navigation

> *communication between navigator and advice provider is non-mutual and in one direction.*

Dieberger (2002) also classifies social navigation as either direct or indirect. It is possible to find other interesting modes of communication, such as, synchronous vs. asynchronous, or intentional vs. unintentional. In the following discussion on direct and indirect social navigation these distinctions will be made when appropriate, but it has to be noted that there is no clear-cut distinction between any of them. To give an example, if a user knows that her movements on the web are recorded as a path, is she intentionally acting as an advice provider to a future unknown navigator?

It is obvious that the proposed definition is rather broad and loaded with interpretation at an individual level. What is an advice provider for one user does not have to be an advice provider for another user. In the end it is the perceived advice provider and her actions that decides if the style of navigation can be called social.

2.3.2 DIRECT SOCIAL NAVIGATION

Direct social navigation is characterized by a mutual communication between user and advice provider. A user can ask questions like "Where am I?" or "Where can I find location X?". The advice provider answers the user and, perhaps more importantly, can ask the user to clarify her questions. An advice provider can, thus, help clarify a user's goals or even change them. When a user is uncertain of where she wants to go the advice provider can support her in formulating a destination. As explained earlier (see Section 2.2) the navigational activity has several components, two of them being to recognize when the destination has been reached and also to choose the correct route to that destination.

Direct social navigation is often synchronous (real-time based), but there are cases when it is based on asynchronous communication. A specific reply to a question in an email list is regarded as an example of direct social navigation, since the communication is still mutual although not synchronous.

COMMUNICATION BETWEEN A USER AND HUMAN ADVICE PROVIDER

The most common way of doing direct social navigation is to use human advice providers. This type of social navigation is suitable for multi-user environments such as the web or some other single-user environment that is connected to a multi-user environment. For example, it is possible to imagine a scenario where a user is stuck in her spreadsheet. The user, instead of consulting the spreadsheet's online manual, sends out a request for help to a pool of spreadsheet experts. An expert answers the user's request and a real time communication begins to navigate the user to her goal. In this scenario it is also likely that the expert will help the user to reformulate her goal, as it is often the case that we have vague ideas of what we are trying to accomplish.

When a user consults an advice provider the user needs some sort of guarantee that the advice provider is the expert that she claims to be. This may seem like a minor issue, but when asking for navigational advice concerning pharmaceuticals this becomes crucial (Impicciatore, 1997). One way to solve this is to adopt the MUD solution by introducing players and wizards. Wizards are expert MUD players that can only be granted their wizard status by other wizards. Assuming that a wizard does her job properly, a user can be sure of getting accurate answers from her. There are basically two ways of identifying experts. A system can either test the expert's knowledge in an automated fashion as done in the SATELIT system (Akoulchina and Ganascia, 1997) or other experts can grant expert status, e.g. becoming a wizard in a MUD or an official guide in the PowWow system. Which one is preferable depends on the domain. In complex and large domains it may be difficult to find good automated ways of identifying experts.

In the PowWow system[2] there are a number of official guides that help novices. The guides can be found in certain communities[3]. In the transcript shown in Figure 2.1, the user Swede could not figure out how to view a personal profile. The PowWow online help did not give him much assistance, so he consulted one of the online guides. Swede is clearly trying to move from one location to another. He wants to move from his current location, to the location from where he can see a user's personal profile. In order to navigate the information space – in this case the PowWow personal communicator – he uses GUARDIAN. He can at any time consult him to get help on where he is in the information space. He also uses GUARDIAN to find his destination and, more interestingly, to change his goals. His intention was at first to view another person's personal profile, but he ended up installing a new version of PowWow. His new goal is of course only a means to the higher end of viewing a person's profile.

Another example of this direct communication between two users is to navigate the information space together (see Figure 2.2). User Freddie asks Swede if he knows any interesting web sites. Freddie uses Swede as a means to not getting lost in the information space, Swede can be thought of as the co-navigator on a ship. User Freddie explores the information space (the web) with help of his advice provider that takes him to interesting places. The situation is somewhat similar to exploring an archipelago with a compass and chart without having any predefined destination.

Artificial agents are very good at keeping a history record on what is happening in the information space. They are, however, equally poor in helping a user with problems outside their information space or knowledge domain. Humans are superior to artificial agents on this account. Even if they do not have complete knowledge

---

[2] A former chat system that shut down in the year 2000.

[3] A PowWow community is a chat room that discusses a certain topic, for example, the PowWow community "New PowWow Users".

of a certain domain they can make qualified guesses. In the above example with Swede and GUARDIAN an interesting phenomenon arises. The advice provider knew that another information space (PowWow 3.2) had the information that Swede wanted, which the online help did not know. This may not appear to be a big problem, but with more and more people doing collaborative work and exchanging documents, different versions of the same software can cause huge compatibility problems. Clearly, an artificial agent would not be able to resolve such issues, unless specifically designed for it.

Laila> I see you don't have a profile Swede
"Swede tries to figure out how to view a profile, the help system does not give him any hints"
Swede> Laila, how do view another persons profil
Laila> Ask GUARDIAN he is here to help
Swede> ok
Swede> GUARDIAN how do I view another persons profile
GUARDIAN> Swede, click on the community button, in your personal communicator
Swede> ok
GUARDIAN> now right click on the person whose profile you want to see
Swede> GUARDIAN, nothing happens
GUARDIAN> ok....what version of pw do you have...you can find version number under "About PowWow..." in the "help" menu in your personal communicator

Swede> 3.1, GUARDIAN

GUARDIAN> ok I see...you need version 3.2...go to http://www.tribal.com/download to get it

Swede> thanks!

Figure 2.1: Direct social navigation in PowWow.

Freddie> Swede do you know any cool sites
Swede > follow me and I'll show you
"Swede sends a request to Freddie to join in a web cruise"
"Freddie accepts"
"Swede loads a HTML-page in his browser"
Swede> is anything happening with your browser
Freddie> yes it loaded a HTML-page

Figure 2.2: Two users navigating a space together.

Example 1: TheHighMage says, "Julia, I'm bored. Where should I go?"
Julia says, "You should go see gilded door, TheHighMage."
Julia says, "From here, go present, out, north, n, w, n, up, up, 7, up, up, s."

Example 2: You whisper, "where are we?" to Julia
Julia whispers, "We are in "The Inter Nexus""

Example 3: You say, "julia, where is leira"
Julia says, "Leira was in the Inter Temporal Chat Lounge about 6 weeks ago."

Figure 2.3: Conversation with Julia (Foner, 1997).

UTILIZING SEVERAL ADVICE PROVIDERS

It is seldom the case that one advice provider possesses all knowledge. One person knows the best pubs, while another knows the best way to get to the airport. A navigator (or user) often makes a selection between the possible advice providers based on certain criteria. We ask the people we know for advice, we go to the doctor's office when we need medical advice, we try not to disturb other people too much with "stupid questions", and so on. In the real world the selection process works satisfactorily, but in the virtual world it becomes more complicated. We do not see the person at the other end and we often have no means to confirm that the advice provider is who she says she is. It is obvious that in a multi-user setting (such as the web) it would be a tremendous gain if a navigator could find the right advice provider for a given problem. It would serve both navigators and advice providers – the navigator gets the best navigational advice and the burden on advice providers is distributed. Let us revisit the example (Swede, Laila and GUARDIAN) above and look at it from a slightly different perspective. It could be argued that Swede did not use one advice provider, but in fact two. First he asked Laila for navigational advice and then GUARDIAN who was better equipped to help him.

The issue is, of course, how the navigator finds the right advice provider for the situation at hand. One way to do it is to ask a person who redirects you to the proper domain expert. However, are there no smoother ways of doing it? It should be possible to build the knowledge about advice providers into the actual system or space that a user is in. In PowWow, for example, online guides could be accessible directly from the help system, instead of from special chat rooms. Another simple approach is to distribute a user question to every advice provider. Phelps (Collins et al., 1997) and Answer Garden (Ackerman and Malone, 1990; Ackerman McDonald, 1996) are two systems that utilize several advice providers.

Phelps is a system that supports both direct and indirect social navigation. The idea is that training should take place during work. The system is built to support a set of (static) tasks that employees perform. Each task contains a set of sub-tasks (a checklist) that an employee steps through when performing the task. If an employee gets stuck on one of the tasks, the system offers help through a list of potential peer-helpers. They are other users of the system who have successfully solved the task. The employee can contact one of the persons on the list over the phone or in a CSCW system and get "just-in-time workplace training". The selection of peer-helper is based on

- their knowledge of the task at hand,
- whether they are currently logged onto the system,
- how many times they have acted as peer-helpers in the past,
- other critical characteristics such as language.

Phelps has been successfully tested in the Correctional Service of Canada. There are two notable problems with Phelps. The first one is related to the system's use of tasks. Each task is hand-coded with help of ordinary task analysis, a rather time consuming activity. For a discussion on performing task analysis turn to Höök

(1996). This makes Phelps unsuitable for dynamic domains in which tasks or the whole system continuously changes. The second problem is the burden the system places on its users. Each time a user completes a task she has to inform the system. This makes Phelps suitable in domains where people either know each other, or where the gain from actively updating the system is relatively high.

Answer Garden is a tool that supports an organization's collected memory. It does so in two ways. First it records knowledge in the form of retrievable dynamic frequently asked questions (FAQ). Secondly, if a user is unable to find help for a particular question, the question is sent to the appropriate expert. An expert can choose to record questions in the FAQ to be fed back into the loop. Studies show that the human experts were used frequently (Ackerman, 1994), supporting the idea that people can actually help each other. The two things that users felt important were accuracy and speed of answers, but it did not matter where the answers originated.

COMMUNICATION BETWEEN A USER AND AN ARTIFICIAL ADVICE PROVIDER

Direct social navigation with artificial agents already exists in some systems. Several experiments with agents guiding users have been conducted in MUD environments. Julia (Foner, 1997) is a MUD agent or robot that a user can pose questions to. She enjoys a good conversation – her favorite subject is NHL hockey – and when she cannot understand the user she will try to change subject. Julia's main purpose is to keep track of MUD users, but she also has the ability to formulate a user's goals.

In Figure 2.3 three sample discussions with Julia are shown. The first two examples demonstrate two important aspects of her abilities. In the first example the user does not know what to do next, so she asks Julia for help, i.e. a typical example where an agent helps a user in formulating her goal (although more or less randomly). In the second example Julia helps a user to orient her in the environment, which is another important aspect of navigation.

Some interesting properties arise in MUDs or in virtual information spaces in general. Consider the third example in Figure 2.3. It demonstrates one of the major advantages with artificial agents. An artificial agent is extremely good at storing information and retrieving it quickly. Storing large quantities of information and quickly retrieving it is difficult for a human. In dynamic information spaces an artificial agent could be of great help; it is relatively easy for an artificial agent to keep track of the history of events and storing changes.

To be able to support social navigation between a user and artificial agent it is necessary to build some knowledge about the domain and user into the agent:

- The agent has to work in a restricted domain or information space. Take Julia for example, she works in a closed information space with all information about the MUD accessible to her at any given time – the only thing that changes are the users logging in and out.
- It is necessary that the agent has some knowledge of the user. How experienced is the user with the information space, what are her preferences, and what are the

goals she is trying to achieve? The success of MUD agents can be traced back to the fact that there are a very limited number of tasks that users can conduct. No matter how experienced a user is, her tasks will not be that different from a novice user.

- It is important that the user has confidence in the agent. A user's trust in the agent is of course a measurement on the performance of the agent. If the agent always leads the user on the wrong track, the trust in the agent will decrease. This fact makes it important that the artificial agent gives good advice, rather than giving lots of advice. In the information retrieval society this is known as precision and recall (Karlgren, 2000). Precision is a ratio between the number of relevant and irrelevant pieces of advice given, and recall measures the number of relevant pieces of advice compared to the total space of relevant pieces of advice. Amant and Dulberg (1998) showed that agents that help users to navigate should aim for precision rather than recall.

There is no principal objection to navigate socially with an artificial agent. However, it is difficult to create an agent that is able to interact on the same level as the user. It is probably more sensible to use artificial agents in a setting where a user is navigating indirectly with the advice provider, since the agent does not have to engage in "mutual" communication with the user. An advice provider should be regarded as another inhabitant of the space and the more complex the communication between user and advice provider, the burden on the advice provider increases.

With this said we leave the artificial agents, and for the remainder of the thesis concentrate on social navigation based on actions left by human advice providers. To focus on systems that rely on artificial agents will partly entail research in advance machine learning which is not the focus of our research.

2.3.3 INDIRECT SOCIAL NAVIGATION

In indirect social navigation there is no two-way communication between the user and advice provider. This means is that advice providers do not have to be aware of the fact that they are giving advice. It also entails that advice providers can communicate through cumulative information, e.g. by walking down a path in a forest and thus collectively defining it. In this section we look at two common techniques for indirect social navigation: collaborative filtering and history-enriched environments. In history-enriched environments we utilize the fact that people tend to "follow the crowd". In the same way as people follow trails of other in the woods, the intention is to augment objects in virtual spaces with "usage trails". In collaborative filtering the idea is to filter information based on users' qualitative evaluation of it rather than to filter information based on content.

COLLABORATIVE FILTERING: RECOMMENDER SYSTEMS

There is information all around us: newspapers, television, radio, and so on deliver it 24 hours a day. We use various techniques to filter it out the information we need. For example, we allow editors to filter which news items should be in our newspapers. We only listen to the radio stations that we like. We use our friends when we want movie recommendations. We learn by trial and error where to find good information. These are just some examples, but they illustrate that people need to filter the flood of information that is around them and they cannot do that by themselves.

The introduction of the Internet rendered many of the traditional tools for filtering information useless. The large quantities of information available on the web rule out many of the filtering techniques we use in every day life. A common technique to find good information on the web is to use different search engines[4] and hopefully come across relevant information – a tiresome and time-consuming way to find information. In effect, several methods of automatically separating good information from bad information based on personal preferences have evolved. Two such techniques are content based filtering (Malone et al., 1987; Belkin and Croft, 1992) and collaborative or social filtering (Karlgren, 1990; Shardanand and Maes, 1995; Resnick and Varian, 1997). Systems that use collaborative filtering are usually called recommender systems.

In content based filtering information is scanned for specific parts of it that match some criteria, and then rated based on some statistical analysis. Usually keyword-matching techniques are used to filter the information. The user supplies a preference file with keywords that will be matched against documents. For example, if a user is interested in Thai food she could have the filter, on a regular basis, scan a newsgroup for documents that contain the words 'Thai', 'cooking', and 'recipe'. Shardanand and Maes (1995) identify three major drawbacks with content based filtering:

- Items must be parsable, e.g. music files cannot be parsed using content based filtering.
- It is difficult to achieve serendipity. Content-based systems typically do not allow for the discovery of information not sought for.
- Content based filtering disregards qualitative aspects such as genre (Karlgren, 2000). If a poem and a technical text contain the same keywords they are regarded as having the same value for a user.

In addition, the success of the filtering is highly dependent on the keywords the user supplies to the filtering system. In the above example the content-based filter would probably recommend many documents that have nothing to do with Thai cooking, due to the keywords the user sets up. 'Thai', 'cooking', and 'recipe' are words that can be found in many discourses.

To overcome the problems with content based filtering, collaborative filtering has been proposed as an alternative. Collaborative filtering recommends information based on what other people with similar tastes like or dislike (hence the term recommender system). Collaborative filtering is relatively straightforward to implement. All users of the system are connected to a server that stores each user's personal profile. Personal profiles are matched and the system creates clusters of users with similar tastes. Whenever a user comes upon a new piece of information, it will

---

[4] http://www.google.com or http://www.yahoo.com

be recommended based on other similar users' ratings. There are standard techniques for clustering user profiles, see (Breese et al., 1998; Herlocker et al., 1999) for comparisons of different clustering algorithms. If a recommender system is to work some sort of rating of the information pieces has to be done by the users. The more people that rate and the more they rate, the better the system can group users. Ratings are either explicit or implicit. Implicit ratings can be based on the time spent reading an article (the longer the higher rating), while an explicit rating scheme let users score information pieces themselves.

The major benefit with collaborative filtering is that it enables filtering information based on quality – the filtering is not based on content, but on other users' ratings of it. It also allows for recommendations of serendipitous information. A recommender system can find pieces of information that a user has not asked for based on the fact that similar users find that information valuable. However, there are three major problems with recommender systems.

Bootstrapping: In order for a social filtering system to work satisfactory (i.e. to be able to build the personal profiles), it needs input to work with. Bootstrapping (or cold starts) are problematic in two ways: first, recommender systems need a fairly large user group in order to cluster users, and secondly, since new users have a relatively inaccurate profile the system is not able to classify them into the right user group. That is, a recommender system will always have a startup problem for new users.

Ratings and snowball effects: To judge the value of ratings is not all that simple. Users tend to rate information that has received high ratings, creating snowball effects. Miller and colleagues (1997) showed that information with high ratings got rated twice as often as information that was rated low. Furthermore, how do we judge ratings from a user who has created the information? A reasonable assumption is that people give their own information high ratings. Finally, individual ratings can also be valued differently between individual users. It does not have to be the case that the rating '4' has the same qualitative value for two users.

Explaining recommendations: Recommender systems are often anonymous, meaning that a user has little or no clue about the cluster of users that a recommendation is based on. When a user gets a recommendation she does not know which other users this recommendation is based upon. In some situations a natural question would be "Who influences my recommendations?" Knowing who someone is can be crucial. Sometimes it is more important to know that a particular individual likes some piece of information, rather than a group of users.

Let us look at three different recommender systems in some detail. Phoaks (Hill and Terveen, 1996; Terveen et al., 1997) is a collaborative filtering system of Usenet news groups. It extracts URLs from news articles and presents the most frequently mentioned URLs in each newsgroup. As opposed to ordinary collaborate filtering that use ratings (e.g. GroupLens and Ringo), Phoaks does the filtering solely based on automatic analysis of articles, the more a URL is mentioned the higher rating it gets.

Ringo or Firefly[5] (Shardanand and Maes, 1995) is a system for recommending music. Ringo bases its recommendations on how similar users rate music. For example, if a user comes across a new band, the system will either recommend that band or not based on how users with similar tastes as the user have rated the music in the past. In addition to being a social filtering system Ringo also allows individual users to write small reviews about artists and albums as annotations attached to them.

GroupLens (Konstan et al., 1997) is yet another collaborative filtering system for recommending Usenet News articles. It uses explicit filtering to create and group personal profiles. The GroupLens system is connected to a Usenet newsreader and lets people rate individual posting in a newsgroup in terms of quality. The system then ranks articles in terms of their relevance to individual users.

In the GroupLens system it was shown that social filtering indeed enhanced predictions compared to average ratings on each article. However, Konstan and colleagues (1997) note that it is much harder to use collaborative filtering on subjective topics such as humor, as opposed to programming. This fact can be traced back to the evaluative attributes of an object (see Section 2.2). What is a good joke is to a greater extent more subjective than what constitutes, for example, good programming practice.

[5] Firefly was the commercial version of firefly.

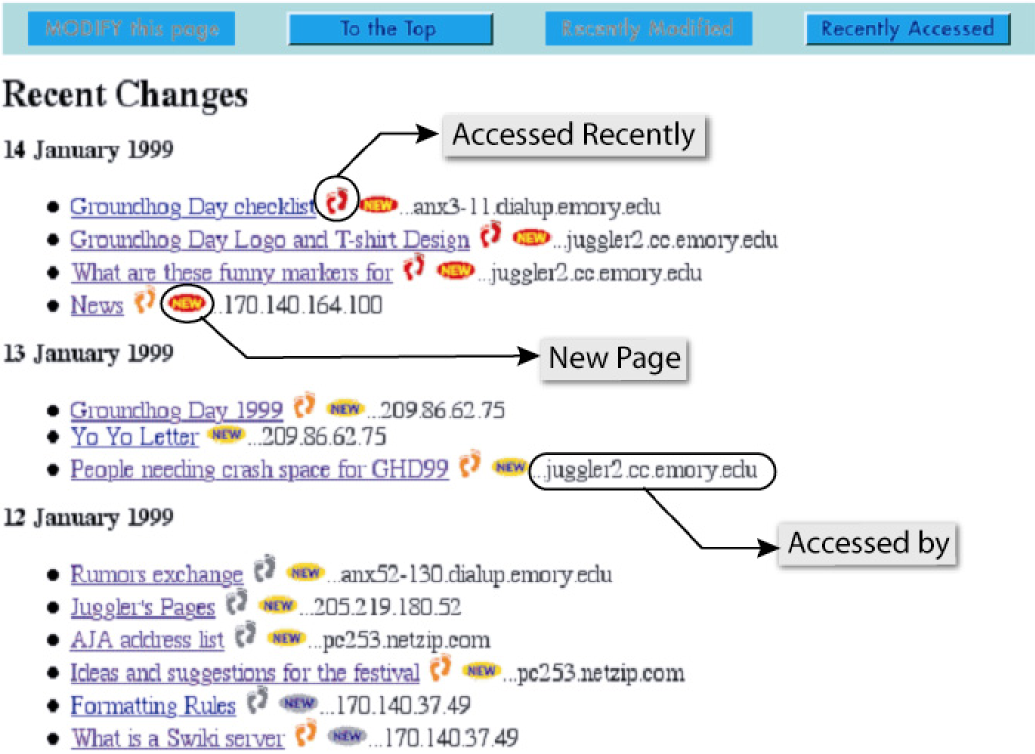

Figure 2.4: The CoWeb (Swiki) system.

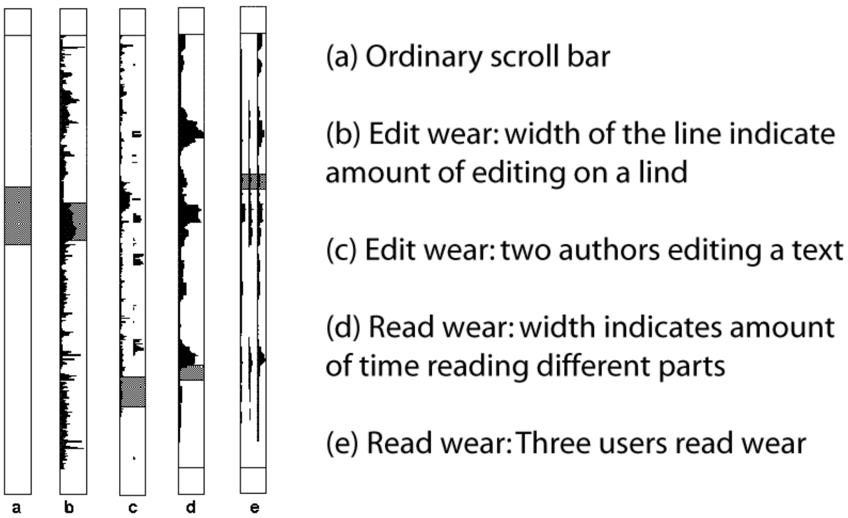

Figure 2.5: Examples of read wear and edit wear (Hill et al., 1992).

History-enriched environments

What other people have done in the past tell us something about how to navigate an information space. If a person gets lost in the woods and comes upon a trail it is a good idea to follow that trail. History-enriched environments make use of the same idea; the more people that take a certain path through the information space the better that path is. History-enriched environments were first explored by Hill et al. (1992). They did not explicitly name them history-enriched environments, but talked about *wear* on objects, e.g. edit wear and read wear.

Wexelblat (1999) defines a framework for history-enriched environments. It serves as a guide for designers of history-enriched environments, and to describe the properties of such environments. Wexelblat identifies six properties:

- Proxemic versus Distemic. A *proxemic* space is transparent to a user in the sense that signs, landmarks, semantic relations, and so on can be easily understood. On the other hand a *distemic* space is opaque to a user. In a distemic space a user lacks the necessary means (e.g. background knowledge) to interpret the space. As an example, the "red footsteps" in the CoWeb system (see Figure 2.4) would be distemic to a user who does not know that they mean 'recently accessed'.
- Active versus Passive. Interaction history is often passive; a user leaves trails without thinking about it. Conversely, when a user updates her browser "hot list" she actively records interaction history.
- Rate/Form of Change. History-enriched environments need a way to deal with the fact that history changes. It can both fade out (e.g. an object has not been used for a long time) or it can accumulate. For example, it is not enough to establish that a web site has been visited many times: a system should also be able to point out which pages are the frequently visited ones.
- Degree of Permeation. To what extent is the interaction history a part of the environment it reflects? A torn book would have a high degree of permeation since it is possible to see from the book itself that it has been used over and over.
- Personal versus Social. Interaction history can be personal (web browser history) or social, i.e. collected interaction history from a group of users (as done in recommender systems).
- Kind of Information. It is possible to collect all sorts of history information: "What was done?"; "Who did it?"; or "Why was it done?". The kind of information to record is highly dependent on the task and domain.

History-enriched environments have been used in MUDs to guide (novice) players in the right direction (Dieberger, 1996). For example, when a player enters a room in a MUD it could provide the following information:

> "*This is the entrance to the mud world. On your right hand you can see a door that doesn't seem to have been used in ages. In front of you see a big portal that several people have stepped through in the past.*"

Apparently the intention is to lead a user in a certain direction based on what other people have done in the past. It is interesting to note that several people do not actually have to use the portal. The important thing is that the user gets the impression that this is the case. Thus, a designer could use strong social indicators to guide newcomers to a specific location.

The CoWeb system (Dieberger and Höök, 1999) records whenever a user of the system enters and leaves a web page. The CoWeb marks this by putting icons next to links. The icons change color over time, so if a link has not been accessed recently, the icon changes color gradually from red to gray, to dinosaur, to no icon at all. CoWeb also highlights new links (see Figure 2.4). In addition to the annotation of links CoWeb also allows users to examine the history of a web page. Users can get detailed information on who has recently accessed a page and who has recently modified it. One problem with history-enriched environments is that they have snowball effects. Considering the CoWeb for example, if only one user has visited a page, it will be marked as 'hot' even though it may be the wrong path to take. If the next user chooses to follow the indicated path the snowball gets bigger.

In Hill and Holland's implementation of Edit Wear and Read Wear (1992) a technique called *attribute-mapped* scrollbars are used. This means that wear is marked in a scrollbar next to the document relative to line position. In the case of Edit Wear it means marking authoring history (i.e. how much that specific line has been edited), and in the case of Read Wear it means marking how many times a particular section of a document has been read (see the examples in Figure 2.5).

2.4 COMPARING SOCIAL NAVIGATION TO OTHER ACTIVITIES

Social navigation is *navigation that is conceptually understood as driven by the actions from an advice provider*. It is clear that the definition is not enough to make a clear distinction between social navigation and other activities such as general information gathering. A framework is lacking that classifies a system as supporting social navigation or not. It is possible to view social navigation as an activity that gradually moves towards four other activities. That is, social navigation can either move towards:

- General social interaction – the activity looks more like social interaction than navigation.
- Navigation – social cues that drive navigation are so inherent in the environment that they can be regarded as any other navigational aid.
- CSCW – the activity is more of a collaborative effort to solve a (navigational) task
- Information retrieval – what is the difference between navigating information space and searching for information?

Rather than using an axiomatic definition of social navigation it is more useful to provide prototypic cases in terms of the differences and similarities between social navigation and the four activities outlined above. It is, in the end, the navigator who perceives and interprets the space she is in.

Social navigation vs. navigation. Two additional properties are needed to describe the phenomena we aim to capture: *personalization* and *dynamism*. Two examples illustrate their importance:

> *Walking down a path in a forest is social navigation, but walking down a road in a city is not.*

> *Talking to a person at the airport help desk that explains how to find the baggage claim is social navigation, but reading a sign with (more or less) the exact same message is not.*

Both methods in these examples seem to convey the same navigational advice; the difference lies in how advice is given to the navigator. In the first example, the navigator chooses to follow a path based on the fact that other people have walked it. Conversely, walking down a street is not driven by the fact that other people have walked the same street. The street is an intrinsic part of the space. One way to think about this is that social navigation traces are not pre-planned aspects of a space, but rather are "grown" – or created *dynamically* – in a more organic or bottom-up fashion. In this way, social navigation is a closer reflection of what people actually do than it is a result of what designers think people should be doing.

In the second example the navigator gets the impression that the navigational advice is *personalized* to her and the situation allows her to ask for additional information. Also, the advice ceases to exist when the communication between the navigator and advice provider ends. The person at the help desk may have to use different terms, or even speak a different language, to convey the same message to each particular customer. The help-desk worker can also recognize a repeat visitor, and modify the presentation of information based on knowledge that a past attempt has failed.

Another key distinction between social navigation and general navigation is how navigational advice is mediated. In social navigation there is a strong temporal and dynamic aspect. A person chooses to follow a particular path in the forest because she makes the assumption that people have walked it earlier. Forest paths are transient features in the environment; if they are not used they vanish. Their state (how well-worn they are) can indicate how frequently or recently they have been used, which is typically not possible with a road.

Social navigation relies on the way that people occupy and transform spaces, leaving their marks upon them – turning a space into a place in the terminology of Harrison and Dourish (1996).

Social navigation vs. general social interaction. When do we stop navigating and start socializing? Again, it is argued that there is no clear distinction and it is subjective when navigation stops in favor of social interaction. Still, the issue is important, otherwise all kinds of social interaction have to be included in the social navigation metaphor.

In general, indirect social navigation has none or a little element of social interaction to it. The communication between a navigator and advice provider is one-way, which makes it difficult for more complex forms of social interaction.

Direct social navigation will always have some element of social interaction. The difference between direct social navigation and social interaction lies in the prime activity that a user wants to achieve. Both Phelps and PowWow (see Section 2.3.2) support social navigation, but in Phelps the navigational metaphor is stronger and the prime activity of the system is not to socialize as opposed to PowWow, which is a pure chat system.

Social navigation vs. CSCW. Social navigation and CSCW share a lot of properties and problems (Dourish, 2002). A central theme in both activities is the notion of *awareness* of others, and that a user takes advantage of other users. In CSCW the shared workspace is a central theme; a CSCW system should enable a group of users to solve a task collaboratively. Users of CSCW systems are very seldom anonymous. On the other hand, in systems that implement indirect social navigation it is not the case that users collaborate to solve a task, at least not consciously. That is, the advice provider does not have to know that her actions will have a future influence on someone else's navigation.

At the one end there are systems such as history-enriched environments that are radically different from CSCW. At the other end of the scale, systems such as Phelps share a lot of properties with a CSCW system. Perhaps it could be argued that Phelps supports both activities and trying to classify it as either or makes little sense. Nevertheless, it is important to recognize that there are differences (at least methodological) since they will influence both the design and evaluation of a system.

Social navigation vs. Information Retrieval. The fact that navigation can take place in non-Euclidean spaces such as the web, makes it sometimes hard to see the difference between navigation and IR (Information Retrieval). Is a person navigating or searching for information when she types a question into a search engine? Traditionally, IR is viewed as the activity of retrieving specific documents that match a certain query from a corpus of indexed documents (Baeza-Yates and Ribeiro-Neto, 1999). In a typical IR system the user is presumed to have a well-defined information need that can be matched against some information repository. In navigation, on the other hand, the need (or destination) may not be clear to users.

Whether or not a search in a search engine is navigation or not, depends on a user's intentions. If she knows exactly what she wants and she uses the engine to retrieve that information, we would classify it as information retrieval. Conversely, a user that uses a search engine in order to clarify her goal, or get a better understanding of some subject, is navigating.

In general, we argue that navigation always has an element of information retrieval to it. In navigation the navigator continuously has to retrieve information that tells her where she is, where to turn, when the goal has been reached, and so on.

3 DESIGNING A GENERAL TOOL FOR SOCIAL NAVIGATION

It is possible to imagine that social navigation consists of a set of well-defined functions that cleanly map onto different situations and domains. In Chapter 2 we demonstrated that this is not the case. There are numerous dimensions to social navigation and ways to support it. In some situations a direct social navigation approach is appropriate while in another history-enriched environments should be used. In this chapter we look at a set of design principles that we believe are important when designing systems that support social navigation: *awareness* of people, *trust* in advice providers, *privacy* for advice providers *appropriateness* of social navigation, and *integration*.

The proposed principles and general findings on both navigation and social navigation serve as the foundation when, in Section 3.3, we propose a design for the Social Navigator toolkit. Since the Social Navigator is a designer's tool it has to meet two additional requirements:

- It has to be accessible from a number of platforms.
- It has to be accessible from several types of end-user interfaces.

The design and implementation of the toolkit will be discussed in detail. We will also give examples of how it can be used. In this way the chapter serves as a practical introduction for designers. Later, in Chapter 4, we revisit the principles from the designer's perspective and use the Social Navigator to design an online food store that uses social navigation as its fundamental mechanism for navigation.

3.1 SOCIAL NAVIGATION AND IMPLICATIONS FOR DESIGN

There are several factors that influence social navigation. As discussed in the previous chapter advice providers communicate actions to advice seekers. Communication of actions can, basically, happen in two ways; the advice provider communicates her action directly or indirectly to the advice seeker. Secondly, actions can be communicated unintentionally or intentionally. Thirdly, there has to be some balance between giving advice and receiving advice. Finally, since social navigation relies on people communicating actions, the issues of trust and privacy are important. These are factors that will influence the design of a system that supports social navigation. To successfully implement social navigation a designer needs to investigate

- the role of the advice provider: "Is it important that she is trusted?"; "How should actions be communicated between advice providers and advice seekers?"; "Are there any privacy issues concerning advice providers?".
- the role of the advice seeker. Advice seekers need to be aware of advice providers: "How will presence of others and their actions be mediated to advice seekers?".
- how social navigation fits into the overall system. Social navigation is not a tool in itself but a way to support a user in solving a navigational task. A designer needs to investigate how social navigation fits with the overall system.

3.2 DESIGN PRINCIPLES FOR SOCIAL NAVIGATION

Social navigation as a metaphor for navigation is complex and so far the work on social navigation has mainly focused on understanding it. The only strong consensus that can be drawn from our definition of social navigation is that a system that supports it must allow people and their actions to be visible to others. Thus, awareness of people and actions is a crucial component in any social navigation system.

However, some work has been conducted on designing social navigation systems. We have investigated how to design systems that support social navigation (Forsberg et al., 1998). The findings draw upon how people conduct social navigation in the real world (Macaulay, 1998; Munro, 1998; Harper, 2002) and informal analysis of systems that support navigation. Some implications for design are also drawn from a meta-study of adaptive navigation support conducted in (Höök and Svensson, 1999). Wexelblat has investigated the properties of history-enriched environments (Section 2.3.3), which can be used as a basis for design of such systems. The importance of awareness has been investigated in the CSCW community (Ackerman and Starr, 1995). The issues of trust and privacy are raised in (Van House et al., 1998; Dieberger, 2002; Palan and Dourish, 2003) but they are still open and need more work. Thus, there are indications for the important properties that will influence the design and use of systems that support social navigation.

Below we present five design principles that were used when designing the Social Navigator. They can be divided into two categories, (a) principles dealing with advice providers and advice seekers in an information space, and (b) principles that are related to how social navigation should be incorporated into the targeted system. Awareness, trust, and privacy are related to ways of making advice providers accessible to advice seekers. Integration and appropriateness are related to how the social navigation support is mediated in a system and what type of social navigation is best suited for the advice seekers for a particular task and domain.

3.2.1 PRESENCE AND AWARENESS

Social navigation is navigation that is driven by the actions of advice providers. Thus, presence of advice providers and their actions are vital factors in social navigation. It is not only information spaces that mediate presence. Many information artifacts can potentially communicate the presence of advice providers. A torn book cover indicates that someone has read the book or at least interacted with it.

Simply by identifying others' actions we can feel a certain presence of others (Munro, 1998b). Thus, presence does not necessarily mean real-time awareness (that users are in a space at the same time), but it can also be mediated, for example, through trails of other people, as done in most history-enriched environments.

We divide presence into two types: real-time awareness of advice providers and aggregated history of the advice providers' actions. In ICQ[6] only real-time presence is indicated, that is, whenever a user is online this will be indicated to others. The system, however, uses interesting techniques to mediate the notion of presence, indicated by several different forms of real-time awareness. For instance, when a user has been away from his or her computer for some time this will be indicated to others as "user is currently away from her computer".

The torn book cover exemplifies the other type of presence that can be mediated. There are systems that indicate presence by displaying interaction history. A simple hit counter on a web page is a typical example of this. Although extremely simple, it gives some sense that people have visited that particular web page. A more advanced example is the footprints system in which Wexelblat attempts to replicate paths in a forest (Wexelblat, 1999).

3.2.2 TRUST IN THE ADVICE PROVIDER

Trusting an advice provider is important since social navigation is a way to make use of other people and their actions when navigating information spaces. An advice provider could be trusted for several reasons. It could be that an advice seeker knows who left a particular trail in a history-enriched environment or that recommendations are based on users with similar profiles. In direct social navigation it can be vital to know that the advice provider is who she claims to be. Another way of enforcing trust is to have a rating system in which advice providers are rated by advice seeker (compare to epinions.com)

The degree of trust is dependent on the task that a person is currently undertaking and on the information space itself. For example, if a user tries to find information about personal health care, the source and quality of the information is crucial (Impicciatore et al., 1997).

As noted by Macualay (1998) and Harper (2002) the relevance of information is closely related to who provided it and who uses it. Knowing who uses a particular source of information is as important as knowing the reliability of the source. Put in another way, knowing something about the other navigators using the same source of information can be equally important as knowing the advice providers. For example, knowing that the 'experts' use a particular source of information makes the information valuable regardless of its content.

No matter what guarantees a system gives about its advice providers, trust is something that grows over time. It is the "face-to-face" communication between people that creates trust. Systems such as eBay use various 'reputation' mechanisms

[6] Chat system that can be found at http://www.icq.com.

to enhance this process. By allowing users to vote on each other newcomers are able to make a better judgment on whom to trust (Resnick et al., 2000).

At the bottom line it is the navigational advice that is important to the user. If an advice provider gives poor advices the trust relationship is easily broken. A person receiving bad advice from a physician is likely not to trust her any more no matter what her credentials are. This works both ways, if a recommender system provides good recommendations it does not matter where the recommendations stem from, as was shown in (Ackerman, 1994).

3.2.3 PRIVACY FOR THE ADVICE PROVIDER

Potential privacy problems are identified when looking at social navigation from the advice providers' point of view. It might be important for a user to know that her actions are recorded or that a user's action affects the ranking in a recommender system, thus affecting other users who follow. In systems that utilize direct social navigation, such as chatting, virtually any other user can approach advice providers, which can be disturbing at times. The burden of helping others must be proportional to the gain from using the system. This issue is recognized in the Phelps system where potential advice providers are partly ranked depending on how many times they have helped others. If a certain threshold is reached the system will not consider them as potential advice providers (Collins et al., 1997).

Privacy issues have been recognized in several chat systems such as PowWow and ICQ. In these systems users can set their online visibility to the outside world. A user can, for example, set her visibility to "do not disturb", signaling that she does not want to be contacted by others. Whether or not users use visibility flags in these types of systems is something we have not investigated. A few initiatives have been undertaken to deal with privacy in recommender systems. In GroupLens (Konstan et al., 1997) users have pseudo names to protect their real identities.

Since social navigation relies on (more or less) open systems, in the sense that people can see each other and communicate, it is important to provide alternatives for those who do not want to be visible to the rest of the world. These users should not be excluded from social navigation systems, but should be allowed to enter the systems invisibly.

3.2.4 APPROPRIATENESS

We have to realize that social navigation is one of many ways of supporting navigation. Is it then always appropriate to provide socially enhanced information spaces, or can a social space have the opposite effect and make people unwilling to go there? Parts of the web that contains sensitive data (e.g. pornographic material) are certainly more sensitive to social navigation. In those spaces it might not be a good idea to introduce some forms of social navigation such as direct communication between visitors unless anonymity can be guaranteed.

Social navigation is a broad concept and contains several ways of enhancing navigation. In one system indirect social navigation may be appropriate, while direct communication between users may not work at all. Phoaks (Terveen et al., 1997) recommends web pages based on what pages people refer to in Usenet newsgroups. To recommend web pages in a domain containing thousands of discussion groups will probably mean that the recommender system works better for some groups than others. Another example is the GroupLens system, where it was showed that recommendations are more efficient for specialized newsgroups ("Linux Software"), compared to extremely general groups ("humor") where social filtering is of less use (Miller et al., 1997).

To summarize, some domains are not suited for social navigation (or less so). Allowing users to see each other on the Google (http://www.google.com) web site would not necessarily help users find the information they need. The number of people outweighs the benefits of the approach. We also argue that using recommender systems in domains where the reasons for liking or not liking something are many can be complicated. That is, users can for a number of different reasons agree on something being good. Take humor for example, if two users happen to like the same joke it does not necessarily mean that they have the same preferences for humor, and basing recommendations on that single joke could have the effect of recommending bad jokes to both of them.

3.2.5 INTEGRATION

Tools that support social navigation should be an integral part of our everyday tools and not something that sits on the side providing meta-advice. A tool that supports social navigation on the web should either be integrated with a web browser or the web pages themselves (Barret and Maglio, 1997; Marais and Bharat, 1997). Alexa (Figure 3.1) is a navigational aid that is well integrated into the browser it enhances. The social functionality becomes a natural extension to the overall browsing experience.

In an informal evaluation of the online store presented in Section 4.2, different chatting facilities were tried out. Two versions of the system were used: one that used a separate chat window and one that had the chat facility built into the interface. In the latter system an increase in chatting was noticed.

Along the same lines we argue that when integrating social navigation in, for example, a word processor, it should be accessible in the same way as the help system, instead of being a separate tool that the user needs to start on the side and access separately from the rest of the word processor. In general, integration stresses the importance of communicating the knowledge of how the application makes use of social navigation and how the tools for social navigation are related to one another.

3.3 THE PROPOSED SOLUTION

In light of the discussion in the previous section, we propose a tool or framework that supports the creation of systems that utilize social navigation. The tool should not be stand-alone, but be used by designers. The design of the Social Navigator, hereafter named SN, is obviously phased with a number of issues that will affect its functionality and architecture.

The design principles will have a strong impact on the Social Navigator, and they will dictate what sort of functionality it has to support (e.g. it should support awareness of people). Furthermore, the Social Navigator is a toolkit and not an end-user system, which imposes architectural considerations on the final solution. For example, the toolkit should be general enough to be used in conjunction with both HTML based clients and Java based clients.

> **Design choice (integration)**: The Social Navigator toolkit must not specify a fixed end user interface.

The component that handles social navigation should be an integral part of the system it enhances. A designer should be able to plug the SN into virtually any system. It has to work smoothly with a number of user interfaces, ranging from command driven interfaces to HTML based interfaces. The principle of integration thus rules out the possibility that the SN can provide a user interface of its own to the underlying social navigation functionality. The diversity of end-user interfaces makes it impossible to provide a generic user interface to the SN that will work for all of them. It is the designer's responsibility to provide an end-user interface to the SN.

> **Design choice (presence)**: The Social Navigator attaches users to locations and regards movements between locations as users' actions.

A fundamental part of social navigation is to mediate presence of people. By presence we mean real-time awareness of other people and stored user actions. Since the SN does not know what the end user system looks like and what sort of actions a user can perform in it, it is necessary to find a solution that is general in the definition of actions. The SN introduces a scheme based on users and their locations to define actions. A user is always attached to a location and movements between locations are automatically stored. That is, movements between locations represent users' actions and their current locations represent their real time location.

> **Design choice (privacy)**: Flags can be attached to online users that signal their status to the rest of the world.

A system that supports social navigation should respect peoples' need for privacy. In a history-enriched environment advice providers need a way to turn off logging of their actions, and similarly, in a real-time based system a user should have the opportunity to be invisible to the rest of the world. The SN adopts a simple scheme of visibility flags attached to online users. The designer of the end user application decides the meaning of the flags. However, the SN has a special purpose flag that indicates the 'invisible' mode. This mode turns off all logging of a user's actions and she becomes invisible to the outside world.

> **Design choice (trust)**: Flags can be attached to online users to signal trust.

What mechanisms for upholding trust in advice providers can the SN support? The designer can use visibility flags to model trust. Special purpose flags could be used to indicate that particular users are trustworthy or that they are experts. Keep in mind that trust in advice providers is something that builds up over time. If an advice provider continuously gives good advice users will likely trust her.

> **Design choice (appropriateness)**: The SN supports storing of actions, real-time presence, and chatting.

Social navigation can be direct or indirect, whichever is preferred depends on the domain and tasks at hand. Again, the SN is a toolkit that should impose no restrictions on what type of social navigation a designer wants to enhance her system with. As we have seen the SN stores user actions, which is the fundamental building block for history-enriched environments. The SN also supports two ways of direct social navigation: peer-to-peer communication and location based communication between people.

> **Architectural Requirement 1**: The SN should be accessible from a number of platforms.
>
> **Design choice**: The SN adopts a client-server architecture where the end user system accesses the functionality in the SN through a web server.

Obviously a toolkit that intended for designers has its own set of requirements. Being a designer's tool the SN has to be general, allowing it to be used with a range of end-user applications. A designer of an HTML page and a designer of Java applets should both be able to use it. Instead of using an architecture where the SN is actually plugged into the end user system we propose a client-server solution.

The SN is a stand-alone server that the end-user system communicates with when it needs access to the social navigational facilities. It was decided to choose a web server approach with communication based on the HTTP protocol and the main reason for this is that any end user system that can communicate with a web server can access the SN.

> **Architectural Requirement 2**: Different end-user systems should be able to communicate with each other through the SN.
>
> **Design choice**: The SN has a set of functions for communication between end-user systems that are unambiguous in their interpretation.

It is possible to imagine several different interfaces to an underlying information system. The toolkit should be accessible from a variety of interfaces. In reality it means that two users on different modalities (for example, a text based web browser and a stand-alone application) should be able to see and communicate with each other through the SN. In effect, the SN has a set of well-defined functions for communication between clients. These communication channels are accessible to any end-user system that can communicate with the HTTP protocol. For example, as we saw above, the SN has a special flag that marks a user as invisible and it has the same meaning independently of the end-user application.

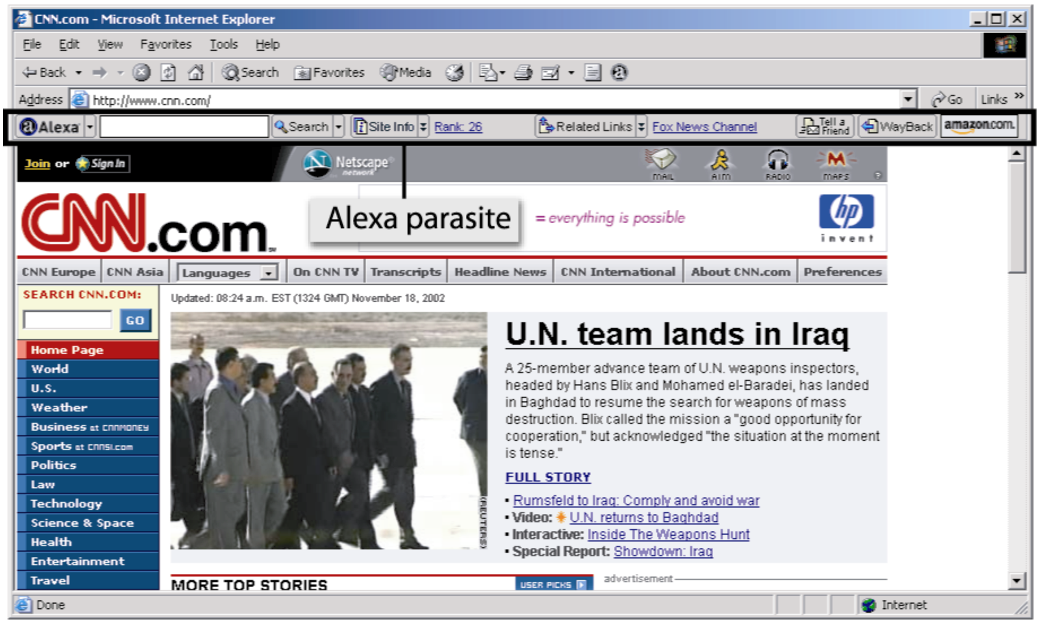

Figure 3.1: The Alexa user interface.

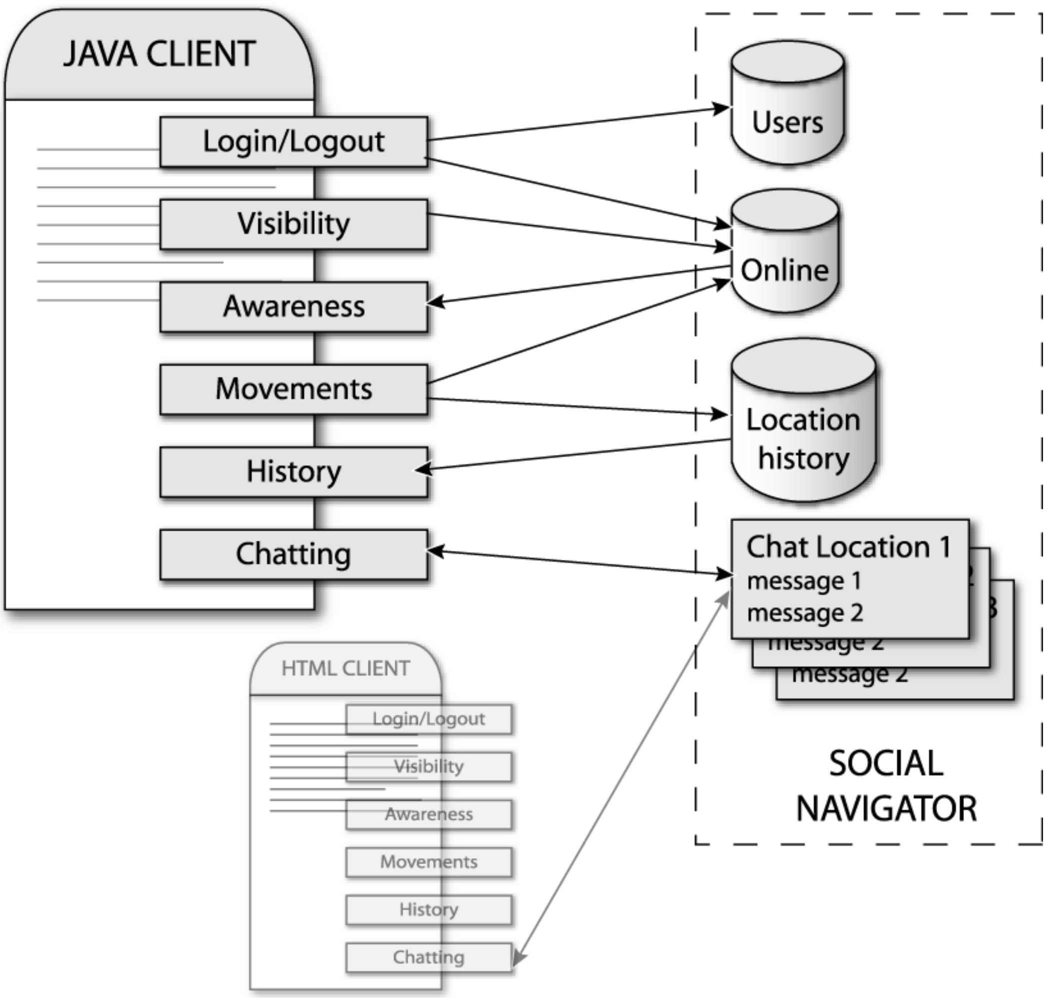

Figure 3.2: Social Navigator architecture.

3.3.1 ARCHITECTURAL OVERVIEW

Figure 3.2 shows the toolkit's abstracted architecture. At one end we have the clients that access the functionality in the SN. A client can move between locations, set a user's online visibility, and send chat messages to the SN. With this basic model clients can retrieve information about where users have been (i.e. history information in Figure 3.2), get chat messages on different locations, and get information about the users that are currently online (i.e. awareness).

From Figure 3.2 it is also clear that clients can use different end-user interfaces. In the figure there are two clients currently connected to the SN: one using a Java interface and one using an HTML based interface. For both clients communication with the SN is the same, the only difference would be the way they handle user interaction and what functionality they utilize in the SN.

No matter how well the Social Navigator is designed, it remains a designer's tool and not an end-user application. In the end it is up to a designer to use it in the way it is intended. Without an understanding of social navigation and especially the design principles that the tool is based upon, it will not serve its purpose. To emphasize this point, to understand the Social Navigator, a designer has to understand social navigation.

3.4 THE SOCIAL NAVIGATOR

Starting from the architectural requirements we describe the Social Navigator architecture as a whole. We examine its different components and discuss how they are related to each other. The main focus will, however, lie on how the design principles are implemented, i.e. the actual functionality of the Social Navigator. While the Social Navigator should be accessible through virtually any interface, we have decided to implement a default Java interface.

The complexity of designing a toolkit that can be used within the range of end-user applications we want to support rules out any architecture that is directly plugged into an end-user application. Such a solution would be language dependent. End-user applications would have to be developed in the same programming language as the SN to access the functionality it provides. The other possibility would be to allow designers to access the SN through an API[7] (independently of language), which entails that end-user applications have to be implemented on the same platform as the SN. However, we do not want to restrict the SN to either a specific platform or language.

We therefore focus on a client-server solution. In short this means that a designer using the SN does not incorporate it in her system *per se*, but rather communicates with a server that handles users, locations and actions. The social navigation functionality is moved out from the end-user application to a server and the client designer need only to request information from the server and present it to the user.

[7] Application Programmer Interface. An interface to a program that specifies a set of functions that another program can use in order to access its functionality.

Instead of developing a new server with its own protocols for communication we chose to base the architecture on a HTTP web server. This has a number of advantages over developing a server from scratch. First, web servers can be found for almost any platform, i.e. platform independence is achieved. Second, a number of programming languages have built-in support for the protocol. And last, a web server can be accessed from within a web browser, which was a requirement for the Social Navigator.

To illustrate how the SN works let us open the discussion with a scenario where an end-user application requests all users that are known to the SN (each step can be followed in Figure 3.3):

1. A Client (end-user application) sends a request to the web server to open a session with the SN: *http://server/sn/LoginUser?login=loginName&pass=password&instance=defaultNavigator.*
2. The web server passes the login request to the Social Navigator, which calls the appropriate navigator instance that logs in the user and sends back a session id to the client (used for further communication)
3. The client next sends a request to the web server to retrieve all users: *http://server/sn/GetUsers*. Based on the session information stored the GetUsers servlet calls the correct navigator instance that returns all known users for that particular instance.

In Figure 3.3 we see that the Social Navigator consists of three components: a web server, the Social Navigator Servlets, and a number of navigator instances. The web server handles communication with the client, the servlets act as proxies between the client and navigator, and the navigator defines the core functionality.

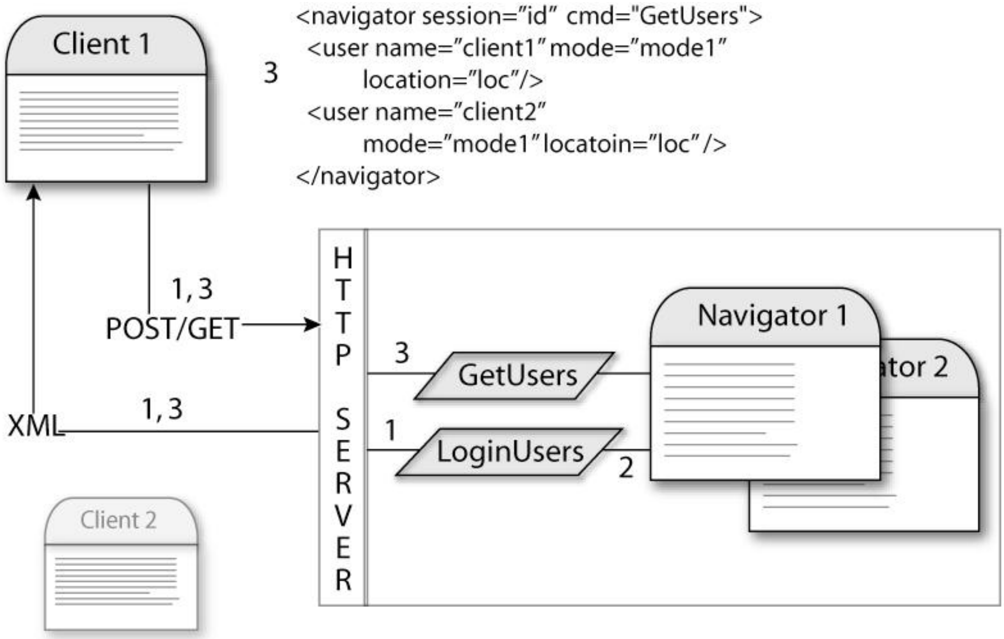

Figure 3.3: Communicating with the Social Navigator.

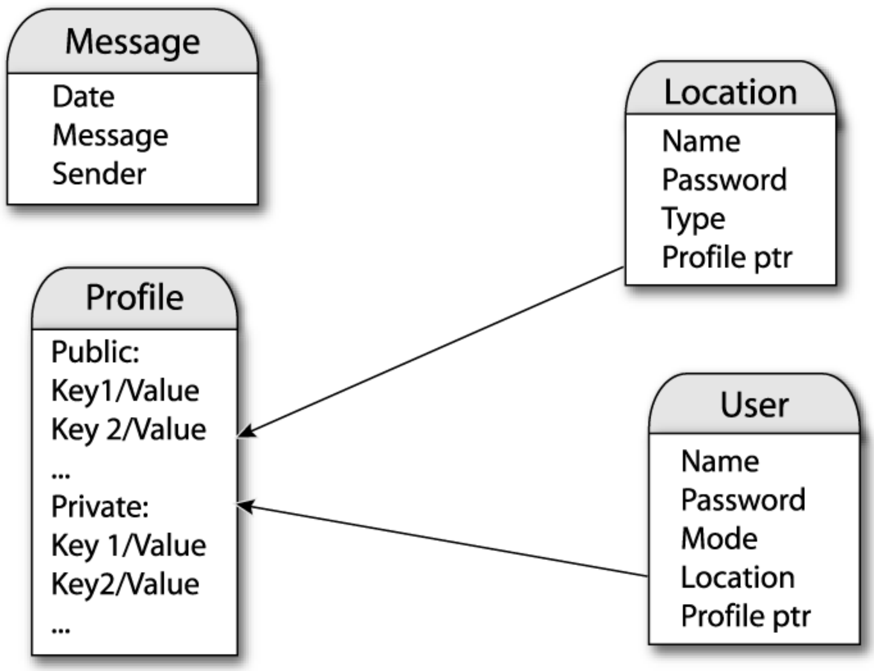

Figure 3.4: Social Navigator objects.

3.4.1 WEB SERVER

There are mainly two ways of using a web server for communication: either a web server with built in support for social navigation is developed or the social navigational information is retrieved from and stored in special purpose modules that are accessed via ordinary HTTP requests. The second solution was chosen for two reasons:

- Flexibility and Portability. A Designer can use any web server she likes as long as it supports the modules that handles the special-purpose HTTP requests to the SN.
- Extendibility. To update the SN does not mean to replace the existing server, but only to provide new SN modules.

A special purpose module can be thought of as a server side program that is executed via HTTP requests. The module accepts input parameters and produces text output using the HTTP protocol. In the previous section, for example, the SN module was requested by calling the getUsers and loginUsers servlets.

There are several ways to implement these server side programs, for example, with CGI scripts[8]. The Social Navigator is implemented as a set of servlets[9]. At start up the web server initializes each servlet and attaches a URL to each of them. Every subsequent client request that starts with that URL (e.g. http://server/-sn/getUsers) is redirected and handled by the corresponding servlet. The main advantage with a servlet approach over other approaches such as CGI scripts is that it is platform independent: any web server that supports servlets can use the same servlet. A designer who wishes to use the SN toolkit plugs it in to her own web server by including the servlets and assigns appropriate URL's to them, and restarts the server.

3.5 NAVIGATOR

At the heart of the Social Navigator toolkit is the navigator, the module that maintains information about users, locations, communication and so on. The other components: the client and servlets are ways to access the functionality the navigator provides. These parts together make up the Social Navigator toolkit. This set up enables a designer to bypass the web-server interface and directly access the navigator. In Java terminology the SN consists of four packages:

se.sics.navigator: core components, i.e. the navigator API

---

[8] Common Gateways Interface. Small server side programs that are used to produce dynamic HTML pages.

[9] Servlets are ordinary Java classes that are loaded into a web server at startup. Each servlet is given a specific URL and whenever the URL is requested the servlet is executed to produce dynamic web content. For an introduction to servlets the reader should turn to http://java.sun.com/products/servlet/ or http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html.

se.sics.navigator.servlet: a set of servlets to handle client requests

se.sics.navigator.client: a client that access the navigator over the http protocol

se.sics.navigator.client.gui: default user interface components for the client

This needs some clarification. On the one hand we have the navigator, an ordinary Java API that is language specific. On the other hand we have the set of servlets, which make up the language independent API. Language independence is achieved through accessing the language dependent API from the servlets. When we present the navigator below the discussion is based on the language dependent Java API.

The navigator itself can be divided into four sections: initialization, user management, location management, and communication. Apart from the initialization step, these map to the fundamental building blocks for social navigation (users, locations, and communication).

3.5.1 INITIALIZATION

There are certain properties a designer would want to set to make the navigator fit the application domain and situation. When a navigator is started it reads an XML based file that contains a number of initialization parameters. The values that can be modified are the following:

- **<admin-passwords-encrypted/>** A navigator has one or more administrators that have full access to it. If this flag is set the passwords will be encrypted.
- **<user-passwords-encrypted/>** If this flag is set user passwords will be encrypted.
- **<location-passwords-encrypted/>** If this flag is set location passwords will be encrypted.
- **<collect-user-history mode="public"/>** Indicates if historic information about users should be saved. If the mode is set to public it is visible to others
- <**collect-location-history />** Indicates if historic information for locations should be saved.
- **<database directory="navigator1"/>** The name of the directory in which the navigator stores its data.
- **<admins file="file"/>** The name of the administrator username and password file.
- **<location name="default" password="default"/>** Automatically create the given location if does not exist.
- **<publicKey name="default" key="description" value="Default"/>** Attaches a property to the location (see profiles below).

The main purpose in the initialization step is to specify the level of security and bootstrap the navigator instance with a number of locations. A navigator requires at least one location to function properly.

3.5.2 Containers

Users

For each user, the Navigator stores the following: a name, a password, a mode, a location and pointer to a profile with additional data (see Figure 3.4). The profile defines a private and public section of key/value pairs, with public keys being visible to any user in the system. The private keys are, on the other hand, only visible for the user. The Navigator stores all this information in a BPlus Tree[10] with the profile stored in a separate file with variable-length records. It is only a minimal amount of information stored in each user. The location and mode allow the user manager to filter users on locations and with different modes. This follows from the design specification that users are always at a location with a certain mode (for example visible or invisible). From this it also follows that a Navigator must contain at least one location (see the initialization step above). Other information such as email could have been included as well. However, it should be the designer's decision to include the type of social information that is needed.

Locations

Locations are very similar to users; they, for instance, share the same profile management. Each location has a name, password, type, and pointer to a profile. The type is somewhat akin to modes in users. They allow a designer to further control locations, for instance, by grouping related locations.

Messages

The last thing that needs to be modeled is communication. The navigator has a very simple view on messages, they consists of a sender, a text, and a timestamp. Messages are collected into buckets that are identified by a text string. To allow for direct communication at a given location the navigator creates a new bucket with the location's name. Similarly, to handle personal communication, the Navigator will automatically create a message bucket when a users logs in. The bucket is identified by the user's login name.

3.5.3 Supporting indirect social navigation

The navigator supports indirect social navigation in either of two ways: awareness of others and recording of users' actions. In the following discussion it is assumed that the navigator is initialized with full historic information (defined in the initialization procedure).

Online Awareness

The following functions are used to promote awareness:

[10] BPlus Trees are data structures well suited for fast disc based access. Our implementation is freely available from http://www.sics.se/humle/software.php

setOnlineLocation(login, password, location, checkLocation, updateOnDisc)
editUser(login, password, newPassword, newMode, newLoc)
Iterator getOnlineUsers()
Iterator getOnlineUsers(modes[])
Iterator getOnlineUsers(locations[])
Iterator getOnlineUsers(modes[], locations[])
Iterator getUsers()

A location is attached to a user as soon as she logs in to the navigator, and is changed by calling *setOnlineLocation*. The navigator has no pre-defined set of locations, however, at least one location must exist in the system to make it work. The checkLocation parameter tells the navigator that the location has to exist and updateOnDisc makes the navigator immediately update the user on disc (to keep the database in a synchronized state).

The user also has a mode, which indicates her online status to the rest of the world. Visibility or mode is set with *editUser* where newMode is an integer describing a user's status. Apart from the special mode visible, it is up to the designer to define the semantics of any included visibility flags. For example, a designer of an ICQ-like clone could define the additional flags: "available for chatting", "do not disturb", and "be right back". The reason for defining a default invisibility flag is to guarantee that users can always turn invisible. This is the only mode that the navigator treats specially by not giving any information about users that have it set.

With this simple setup it is possible to promote awareness in several ways. First a client can request every online user with *getOnlineUsers,* which returns an Iterator over all users that are currently online. Three more functions are defined for iterating over online users by filtering them either on modes or locations. The mode and location filters can also be combined to create a "boolean-and" filter. For example, *getOnlineUsers({location1, locaton2}, {mode1, mode2})* would return those users who are either at location 1 or 2 with mode 1 or 2. Lastly, *getUsers* returns all users known to the navigator instance, both online and offline users. The filtering process only works for online users. This design decision was partly based on scalability and performance considerations but it also had a philosophical reason. What does it mean that a user is offline? Does it mean that she is still at the location where she was when she last visited the system? We argue not, and consequently, we choose not to filter them on their location. Still a client can retrieve all users (both online and offline) and filter them at the client side.

How can the above information be used for social navigation? Consider the task of enhancing a plain web site with social navigation. Say that individual web pages are mapped to locations in the SN and that the user mode denotes knowledge rather than visibility. Then, when a user enters the website she logs in to the SN and specifies what type of user she is (e.g. novice, intermediate, expert). When a user changes location within the site the Navigator is informed about this, and updated accordingly with a call to *setOnlineLocation*. The design could also provide an overview map of the web site and at regular intervals call *getOnlineUsers* and display the output in the overview map, as shown in Figure 3.5. This is a very simple, yet realistic, way of providing an overview of where users are within a site. In this scenario a user could also have the opportunity to visualize a group of users in the map, e.g. by requesting *getOnlineUsers("expert")*.

USER PROFILES AND ACTIONS

In summary, the following functions handles profiles in the SN:

addUserProfileKey(login, password, key, value, privateKey)
getUserProfile(login, password)
getLocationHistory(location, password)
getUserHistory(user, password)
doLog(toLog, admin, password)
getLog(admin, password)

Each user has a profile containing both private and public information. A profile can be of any length and contain whatever information a designer needs (as long as it is in text form). The profile is updated with the *addUserProfileKey* function and, conversely, profiles are retrieved with *getUserProfile*. If the password is left out only the public part of the profile will be returned, thus, the navigator interprets this as another user accessing the profile. There are similar functions to update and retrieve profiles connected to locations.

One possible application for user profiles is to fill them with information about what type of interface the end-user application is (e.g. HTML driven or a Java applet), and use this information determine if two clients can establish a connection in some other way than through the SN.

The second application for profiles is for storing and retrieving historic information. Historic information can be logged for both locations and users and is specified at initialization of the navigator. This information is stored as special keys in profiles. For a user the key contains all locations she has visited and the time she visited them. The only difference between this key and other keys in a profile is that it is collected automatically. Thus, clients can manipulate it like an ordinary key. In the same way, the location history will contain information about all users that have visited a location. The location history allows for clustering user movements and gives information about frequently visited paths and locations, turning it into a way to support history-enriched environments. The user history enables people to see how individual users have moved in the space.

Finally, the SN has a global log file that is not attached to individual users. This is the "designers" log file and it is accessed by *doLog* and *getLog* respectively. It is intended as a complement to the automatically collected historic information.

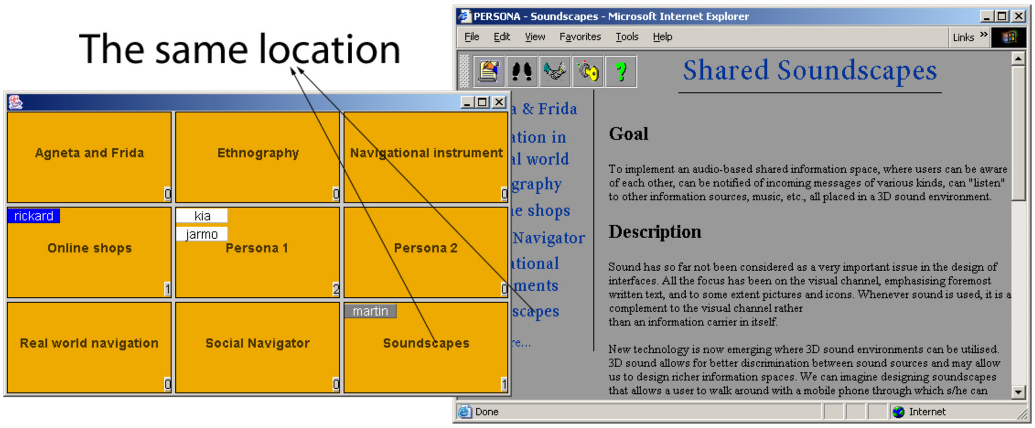

Figure 3.5: A Socially enhanced web site.

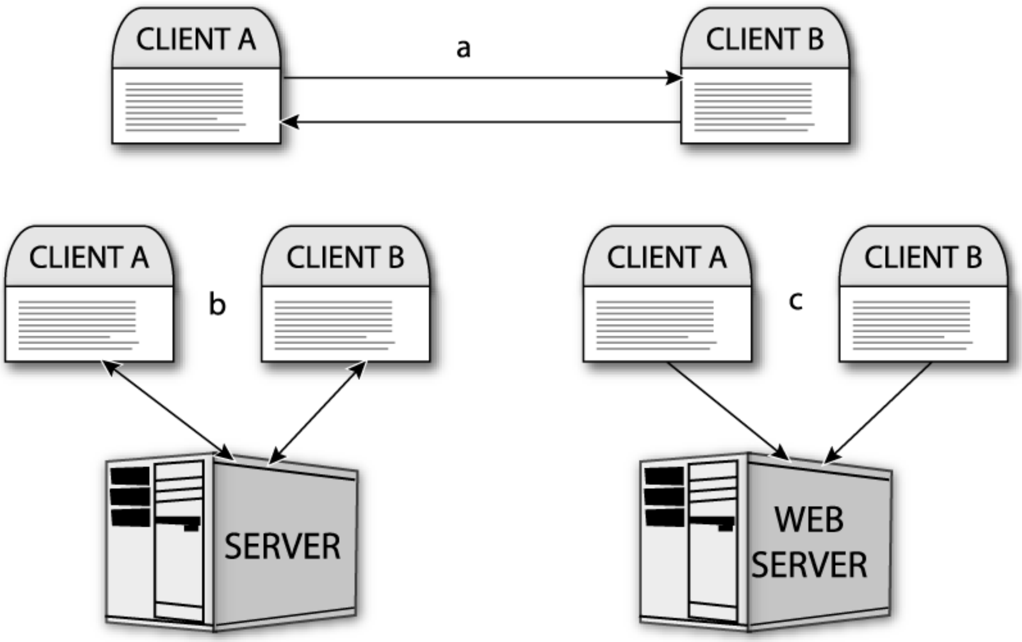

Figure 3.6: Three ways of communication between clients.

```
<profile>
        <key name="client" vlue="recipe-shop"/> //to identify clien
        <key name="host" value="my.computer.com"/>
        <key name="port" value="8888">
<profile>
```

Figure 3.7: An example user profile.

3.5.4 Supporting direct social navigation

The following functions are used to handle messages:

postPersonalMessage(login, password, message, toUser)
getPersonalMessages(login, password)
postMessage(login, password, location, message)
getMessages(login, password, location)
tapLocation(login, password, location)
untapLocation(login, password, location)

The difficult part with the web server approach is to enable direct communication between users. The reason for this is that web servers are not allowed to "push" information to clients, unless the clients request it. HTTP is a stateless protocol that only reacts to requests. This means that true synchronous communication cannot be achieved, since a web server is not allowed to keep an open connection between the client and the server. Normally, synchronous communication is achieved with either a peer-to-peer connection between two clients, or by using a server that keeps open connections to all its clients. Figure 3.6 illustrates this: both in *Peer-to-Peer* and in the *Server* scenario the communication is two-way, while for a web server communication is only one way.

How is direct social navigation handled if synchronous communication channels between clients cannot be set up? In some cases direct social navigation does not have to be synchronous in the strict sense of the word – small time lapses may be permitted for some applications. Through making the clients regularly check the navigator for new messages it is possible to simulate synchronous communication and, obviously, the shorter the interval between client requests, the better the simulation. Thus, it is up to the designer to decide how smooth the synchronous communication should be. The navigator supports direct communication in two ways: personal messages and location messages.

First, a user can send private messages to another user. When a user logs on to the navigator it automatically opens a message container for that user, which can be used to post and read messages. Two functions for personal messages are defined, *getPersonalMessage* and *postPersonalMessage*. For example, if client A wants to send a message to client B (see Figure 3.6c), client A calls the SN with *postPersonalMessage(Client A, Client A' password, Message, Client B)*. Client B later retrieves the message with *getPersonalMessage(Client B, Client B's password)*. The SN returns the message and the name of the user who sent it.

Second, a client can always send messages to a specific location in exactly the same way as sending personal messages. Other clients can request messages from that location and receive the messages sent by the first client, i.e. *getMessages*. The *tapLocation* function sets up a user to listen for messages at a specific location. In this way the SN can always control what messages are sent to what user and that a user cannot retrieve messages sent to given location before she entered it. By default a user automatically taps the location she is currently visiting.

BYPASSING THE SNSERVLET FOR DIRECT COMMUNICATION BETWEEN CLIENTS

While a web server will not allow true synchronous communication, it is possible to use some other communication channel to overcome the problem. It is possible to use the navigator to implement such a 'contact maker' channel between clients. The navigator provides two different ways of achieving this: message passing and through user profiles.

Message Passing. If two end-user applications know that they are of the same type (e.g. only a certain end-user applications will access the SN) and that they can interpret special messages in the same way, it is possible to use direct communication to establish a connection between two clients. For example, client A sends the special message "request chat" through the SN to client B, e.g. *postPersonalMessage(Client A, password, request chat, Client B)*. Client B reads the message and responds, e.g. *postPersonalMessage(Client B, password, accept: my.computer.com:8888, Client A)*. Finally Client A opens a connection on "my.computer.com:8888" and initiates a chat with client B completely separated from the SN.

Profiles. A smoother way is to take advantage of user profiles. In Figure 3.7 a possible user profile is shown. It contains three keys (client, host, and port). In this scenario client A first requests client B's profile - e.g. *getProfile(Client B)* - and recognizes that they are of the same type (recipe-shop). Client A extracts the host name and port number, and opens a connection on "my.computer.com:8888" to initiate a chat with client B.

3.6 SOCIAL NAVIGATOR SERVLET

Every function defined in the navigator has a corresponding servlet. The individual servlets will not be presented in detail. However, we will mention how a client communicates with the SN through a web server and describe the style of communication used.

The main logistics of the architecture lies within the set of servlets that act as proxies between the navigator and client, transforming messages to and from the navigator. The servlets process all clients requests and have a number of commands to meet the needs that were proposed in Section 3.3. The normal way to access the SN is through an HTTP request such as *http://sever/sn/GetUsers*.

3.6.1 SESSIONS

One effect of the stateless http protocol is that all information needed for the web server to process a request has to be passed in every request. For the SN that would mean, for example, that the login and password had to be sent along with every request. However, we have already seen that this is not the case (see Figure 3.3). The way this is handled is that the SN sets up a session on the server whenever a user logs in. The session is given a unique id, and it is this id that the clients send along with subsequent requests. In other words, the client only specifies the login, password, and navigator instance once.

3.6.2 Communication with the client

There are two ways for the SN servlets to communicate output to the client: (1) output can be redirected to other modules before being sent back to the client, or (2) output can be sent unprocessed directly to the client. The main reason for having two methods lies within the capabilities of the client to handle XML. Redirection and additional processing at the server side is necessary for some clients such as HTML based clients. If the output from the SN is not converted to HTML before sending it to the HTML client, it will be presented unparsed.

By default, the SN delivers its output unparsed, and it is then the responsibility of the client to parse and present the output in a suitable way. We do not want the SN to impose any restrictions on how the client treats and integrates data it receives. All output from the SN is in XML form with simple and straightforward semantics. The scheme for output is:

<navigator session=sessionId command=command>
<user, location, or message />
</navigator>

To explain, output from the SN can either be users, locations or messages. Both users and locations may also contain a profile. An XMLConverter is used to transform output to and from the XML format described.

It is possible to imagine a more advanced output format with more semantics. For example, we could use a header that describes the output fields in each record and use an ontology to describe the meaning of the fields. However, it was decided to keep the output as simple as possible. The fact that the servlets only have a limited set of functions and that the output is not interpreted differently in various contexts, makes it possible to send data in this simplified way.

If the end-user application (or client) has no means of parsing and presenting output from the SN we face another problem. Users do not want to be presented with information in the way just described. A designer needs an additional way to tell the SNServlet how to present its output when the client is not able to parse the output on the fly. The way this is done is by redirecting the output to receiving pages such as another servlet that can parse and present the result to the user. The only requirements are that the pages can handle parameters in the same way as CGI-scripts or servlets, and that they reside on the same web server as the SN. Instead of sending the result on the standard HTTP stream the SNServlet redirects the request to the proper page and attaches the result as parameters to that page. This allows the intermediate page to parse the data appropriately and send back, for example, HTML code instead of plain text to the client.

In Figure 3.8 this is demonstrated. The scenario is the same as in Figure 3.3, with the difference that the output from the SNServlet is not sent directly back to the client; it is first formatted by a JSP[11] file (getUsers.jsp), and then sent back to the client.

[11] Java Server Pages (JSP) is another technology for generating dynamic web content. A JSP page is a scriptable HTML page. Scriptable means that a JSP page can contain both ordi-

3.7 COLLABORATIVE AND CONTENT BASED FILTERING

So far we have presented a framework that allows for direct social navigation and a history enriched environment. What is missing is collaborative filtering functionality (i.e. sophisticated ways of allowing users to filter information based on usage history). We could also argue that a social navigation system cannot work properly without ways of navigating with respect to content. In the following chapters, when the online recipe domain is introduced, that type of functionality is indeed introduced.

The Social Navigator framework can be used with two other systems: the Predictor and Searcher toolkits. The Predictor defines collaborative filtering functionality and the Searcher allows for content based filtering. Both frameworks are set up in the same way as the Social Navigator. These frameworks are explained in greater detail in (Cöster, 2002).

The reason for not implementing one framework, which incorporated all types of social navigation defined here, is simply practical. Such a framework would be huge and difficult to manage. Additionally, collaborative filtering is somewhat different from the rest of the functionality in the Social Navigator, and as such, should be kept in a separate framework.

---

nary Java code and HTML code. At runtime the web server compiles JSP pages into servlets. Turn to http://java.sun.com/products/jsp/ for more information.

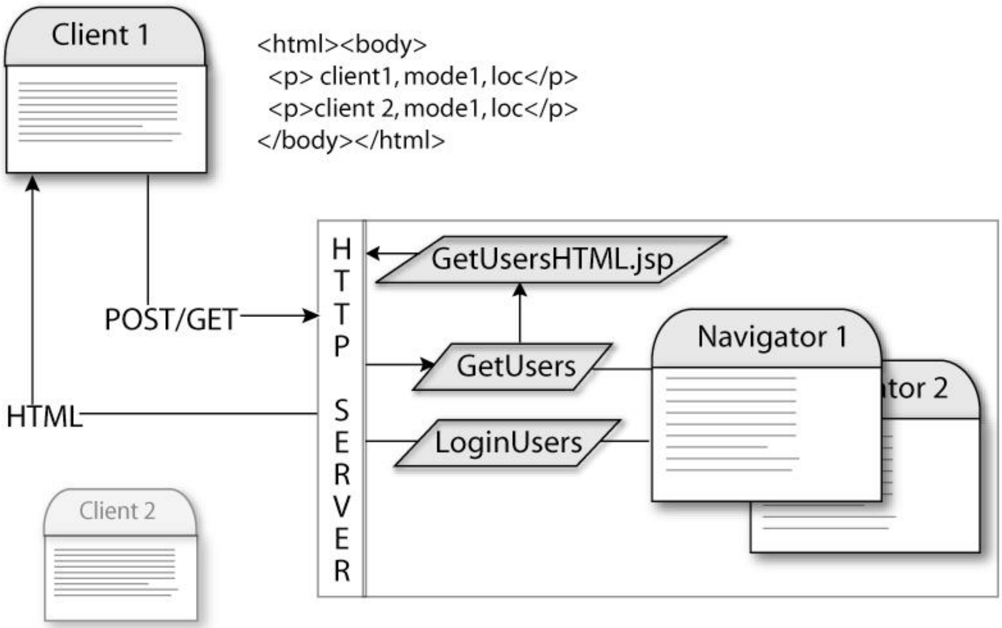

Figure 3.8: Redirecting output from the Social Navigator.

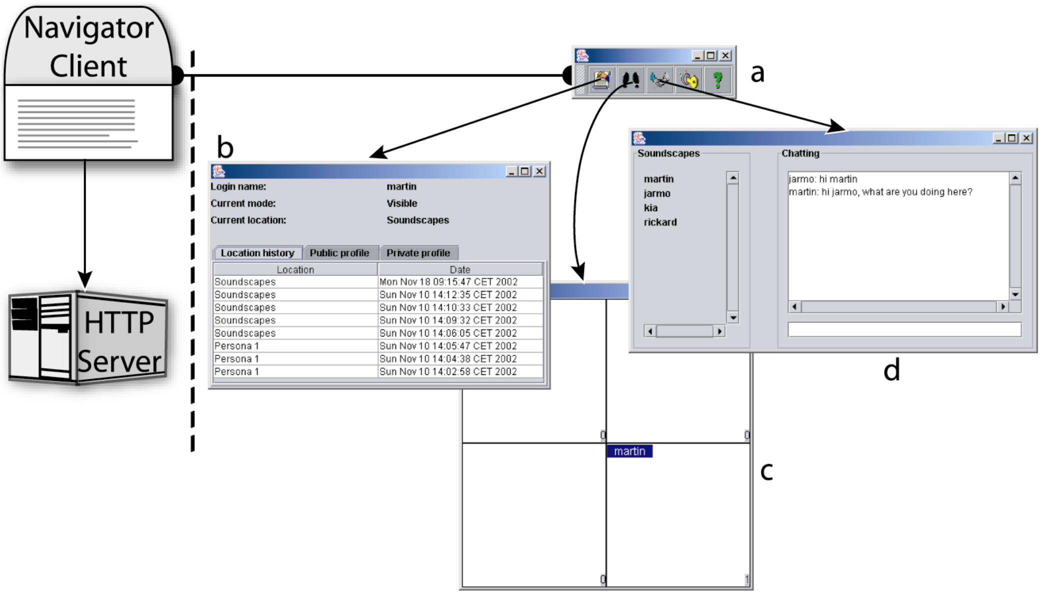

Figure 3.9: Schematic picture of the Java client: (a) SNToolbar, (b) DefaultProfile, (c) DefaultISN, and (d) DefaultDSN.

3.8 DEFAULT JAVA CLIENT

The Social Navigator can be seen as a socially enhanced web server and the responsibility of creating the end-user application is the designers. Nevertheless, providing a default client shell that can be easily modified can reduce the burden on designers. Instead of building an end-user application from scratch, a designer extends a default client. The client handles communication with the SN and provides a user interface where output is presented. The client architecture is split into two parts (see Figure 3.9): an invisible part that handles communication (the NavigatorClient) and four modifiable interface components (widgets a-d in Figure 3.9).

NavigatorClient: The core of the client architecture is the NavigatorClient class. It handles all communication to and from the web server. The NavigatorClient has no user interface and works only as a wrapper. It can be used with any Java application that wants to communicate with the SN. The basic scheme for mapping the SN requests to actual Java methods in the NavigatorClient is as follows:

http://server/sn/servletName?parameter1=param

is translated to the following method in the client

servletName(parameter1).

So for example, the clients's method *createUser(name, password)* would be translated to the HTTP request: *http://pathToServlet/createUser?login=name&password=password.*

An instance of the NavigatorClient handles at most one user at a time, or rather, when a user logs on - *loginUser(name, password, navigator instance)* - it binds that name and password until a *logoutUser()* has been issued. This makes communication easier and less error prone for a designer, e.g. instead of calling *getProfile(name, password)* a designer calls *getProfile().*

SNToolbar: The SNToolbar provides a user interface to the NavigatorClient. It is a floating toolbar that can be put in any Java application or applet. Upon initialization the SNToolbar creates one client object.

In Figure 3.9, there are three buttons highlighted in the toolbar. These are the modules for handling user profiles, indirect social navigation, and direct social navigation (see SNServlet functionality). The 'key' button handles online visibility and user administration (deletion, creation, login, and logout).

The SNToolbar itself does not provide any user interface for presenting user profiles, handling direct social navigation, or indirect social navigation. The reason for this is the principle of integration (see Section 3.2.5). There are a number of ways in which profiles can be used, and the type of social navigation that the end-user application utilizes cannot be determined beforehand. A designer therefore provides custom implementations for this functionality and loads it into the SNToolbar with:

public void setProfile(ProfileInterface profile);
public void setDSN(DSNInterface dsn);
public void setISN(ISNInterface isn);

In turn the SNToolbar attaches each module to the respective button. The only requirement for these modules (or plugins) is that they implement their respective interface – an interface defines a set of methods that the module has to implement. The *ISNInterface*, for example, defines two methods that a module for indirect social navigation has to implement:

```
public interface ISNInterface{
  public void setVisible(boolean b);
  public void setClient(se.sics.sn.client.SNClient client);
}
```

That is, *setVisible* is called when a user presses the button for indirect social navigation in the SNToolbar, and the *setClient* method is used to bind the module to a client object.

DefaultProfile: The architecture provides default implementations for the three modules discussed above, the simplest being the user profile. It displays some user statistics: the login name, mode and location. The profile is split into the public and private parts, with the usage history displayed in a separate tab. The default profile does not allow for any modifications of profiles, thus, providing only read access to profiles.

DefaultISN: More interesting is the default module for indirect social navigation (ISN). There are a number of ways to mediate presence: real-time awareness or aggregated user actions. The module focuses on real-time awareness, and does so by introducing an overview map of the user space. The overview map consists of the locations, and displays users at these locations. When using an overview map to mediate awareness two things are achieved:

- People at the same location can quickly recognize each other. The boundaries of each location are marked with dashed lines (see Figure 3.9c).
- It is possible to use the map as a navigational aid; the map allows for users to navigate by clicking the location (square) they want to go to.

Obviously there are a lot parameters that can potentially be interesting to manipulate in an overview map: "What are the valid locations?", "How many users are allowed in the map?", "How often should the map be updated to reflect changes in the SN?", to name a few. The DefaultISN has several initialization parameters,

```
public OverviewMap(int cols, int xMargin, int yMargin,
  int width, int height,
  int nameWidth, int nameHeight,
  int fontSize, boolean borders, Color borderColor);

public void addLocation(String name, Image background);

public void addName(String location, String user,
  Color background, Color foreground);
```

The overview map may include background images for each location and the designer can specify if the map should have borders around the different locations. The width and height defines the size of each location in the map. The nameWidth and nameHeight together define the max dimension for each user in the map. Locations are then added to the map with *addLocation*. If the background is specified, it will be displayed, otherwise the map creates a simple background by displaying the name of the location. In Figure 3.9c the map contains two columns with four locations.

DefaultDSN: In the default client implementation of direct social navigation chatting can only be done at specific locations and not through sending messages to individual users. When the module is displayed it retrieves the user's current location and opens a communication channel on that location. In the left pane other users at the location are displayed (Figure 3.9d).

The first version of the module was more advanced; it supported both community based chatting and private chats. Through informal tests, this turned out to be complicated for the users. Users could not figure out whom they were talking to, who saw their messages, and so on. It was therefore decided that the default module for direct social navigation should be simple and with only a minimal set of controls.

To close the discussion on the NavigatorClient architecture, a designer who plans to access the toolkit from within a Java application or applet can use the client in several ways. She can either choose to solely use the NavigatorClient wrapper class and build a GUI from scratch or choose to use the various GUI components provided. It should be mentioned that each individual GUI component can be used separately, as long each of them is connected to NavigatorClient. In the first version of the online shop (EFOL) we used the DefaultISN, DefaultDSN, and DefaultProfile without connecting them to the SNToolbar.

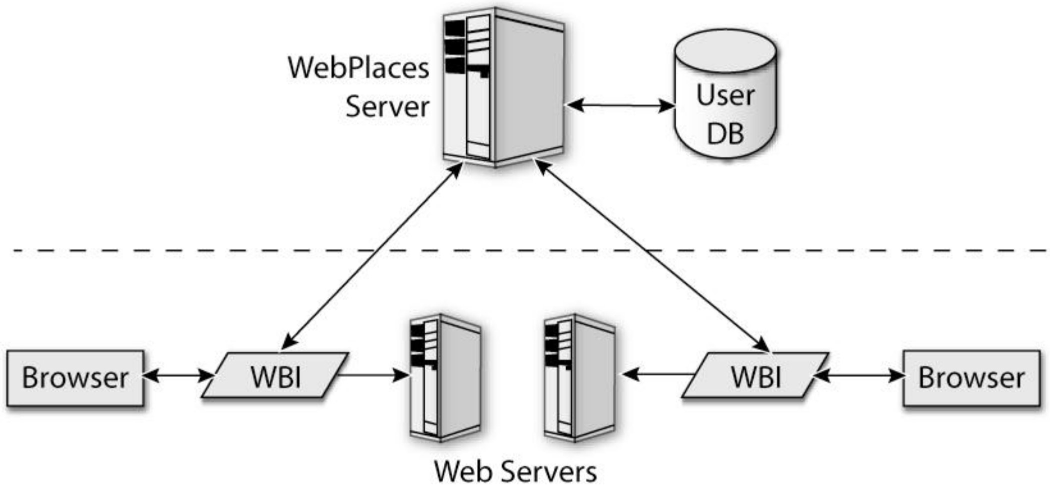

Figure 3.10: WebPlaces architecture.

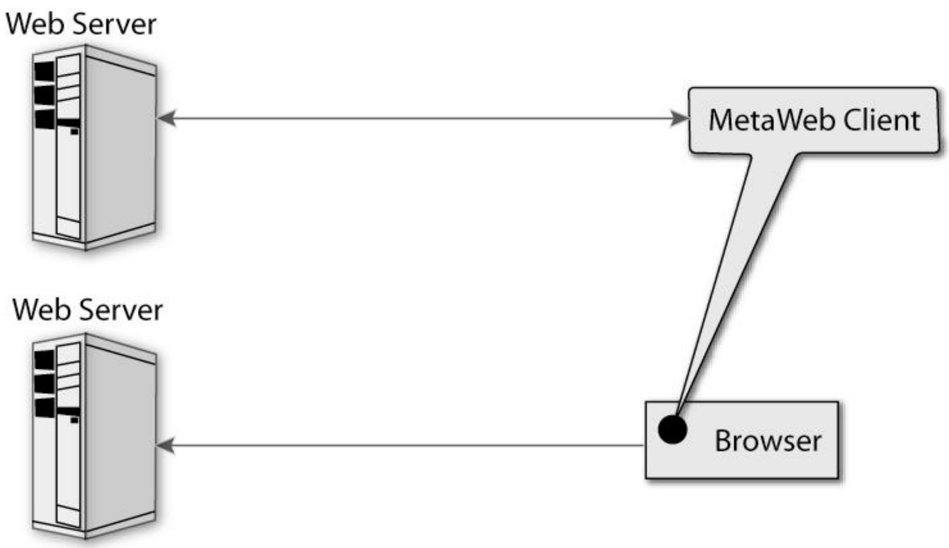

Figure 3.11: MetaWeb architecture.

3.9 A COMPARISON WITH METAWEB AND WEBPLACES

Our design principles were implemented in the Social Navigator. Instead of creating a toolkit we can compare the design principles against existing toolkits to see in what ways those support social navigation. Such comparison is interesting for two reasons: (1) it shows how other toolkits model social navigation, and (2) shows how the SN compares to existing toolkits.

Several initiatives have been undertaken in the CSCW field in order to turn the web into a place for cooperative work (Bentely et al., 1997; Eklund et al., 1999). These are, however, not intended as designer's toolkits but rather as end-user solutions. We have chosen two look at two toolkits: WebPlaces (Maglio and Barret, 1998) and MetaWeb (Trevor et al., 1997). They are both directed at ways of enhancing the web with social navigation, but both can be used in other settings as well. WebPlaces has its origins in social navigation, while MetaWeb is a CSCW system for bringing synchronous communication to the web.

3.9.1 WEBPLACES

As the name implies, WebPlaces (Maglio and Barret, 1999) is a system for turning the web into a place where people meet and interact. Three basic functions are identified:

- WebPlaces allows people that surf the web to see each other.
- WebPlaces displays peoples' activity on the web.
- WebPlaces lets people chat in real time.

Maglio and Barrett (1999) identify three questions as important in their system: How are places recognized? What are the means for awareness of people? How should people be able to communicate?

To answer the first question WebPlaces has two ways of forming communities. The first scheme is akin to the buddy lists found in ICQ, the users themselves define who should be included in the community. The second approach is more complex and utilizes techniques from social filtering. The idea here is that WebPlaces identifies a community based on the way people browse the web. If WebPlaces identifies a commonality in a group of users they will constitute a new web place. Maglio and Barrett identify other possibilities to form communities; one simple scheme would be to form communities based on web pages, i.e. all users currently visiting the same web page form a community.

To make people aware of each other WebPlaces borrow the notion of a social proxy from the Babble system (Erickson et al., 1999). The social proxy is a circle with people represented as dots. Motion of dots towards the center of the circle represents a group discussion, while a person-to-person communication is represented as two dots moving towards each other. To represent user activity - browsing the web – the dot moves around the circle, and conversely, no activity is represented as the dot moving away from the center of the circle. Finally, WebPlaces allows users to communicate in either of two ways: (a) community based chat, and (b) private chats between two users (i.e. whispering).

WebPlaces: design principles

A major issue in the WebPlaces system is to find different ways of mediating presence of other people. What is really interesting with the system is its use of the social proxy. The proxy mediates presence in several ways. First, users see who is online; second, group activity is mediated by user dots moving to the center of the proxy; third, private communication is mediated through two dots moving towards each other; fourth, active users are mediated as moving around in the proxy; and finally, non-active users are mediated as dots moving away from the center of the proxy.

In the current implementation there is no explicit way of mediating trust in advice providers. However, as WebPlaces automatically form communities based on similar users, we argue that it can be seen as an implicit way of signaling trust. Users in a community know that they share common interests, which could enable them to trust each other more than if they were totally anonymous.

WebPlaces does not support privacy in the normal way, i.e. to let people turn invisible. The social proxy is used in an interesting way to support privacy. Users can see when two people are engaged in a private conversation. In our view, the no-activity indicator can also be a way to communicate privacy. The user is engaged in another activity (outside WebPlaces) and does not want to be disturbed.

WebPlaces typically supports one type of social navigation, direct social navigation. The prime usage is to support real-time chat in web based communities. The actions of others are used to mediate activity in the social proxy, which can be a form of indirect social navigation. However, an interesting way to support indirect social navigation in WebPlaces would be to actually connect the activity indicator to specific web pages. That is, it should be possible to see where other people are and where they have been.

A major concern in WebPlaces and the underlying architecture WBI is integration, or as Maglio and Barrett (1998) put it: "In addition, we believe that the most effective way to give users a sense of place *is through the web page itself*". In other words, they recognize that the additional functionality that WebPlaces provide has to be integrated into the system it enhances, i.e. the web browser. WebPlaces is not a separate application but is launched in a separate browser window. While being a part of the web browser, one could argue that putting WebPlaces in a separate window is not the optimal solution in terms of integration. The connection between the user's main task (i.e. browsing the web) and WebPlaces may not be apparent. Just because the dots in the social proxy moves around, a user does not necessarily makes the connection to actual browsing behavior.

WebPlaces architecture: WBI

The WebPlaces system was built using the WBI toolkit (Barret and Maglio, 1997), which is a system for developing intermediates on the web (see Figure 3.10). An intermediate is an entity that lies between the web browser and web server that can alter the content of both requests sent from a browser and responses sent from a web server. A number of modules (created by a designer) are plugged into the WBI to handle requests and responses that meet a certain criteria.

In WBI there are three building blocks that designers use when they create their modules. *Monitors* observe transactions. *Editors* modify requests from the browser or responses from the server. *Generators* produce responses to requests. Finally, a designer has the ability to set up *autonomous agents* that act independently of requests and responses.

The main advantage with this approach is that it will work in situations where the designer has no way of altering the server or the client. The WBI actually sits between them and modifies the streams sent back and forth between the server and client. For instance, it is possible to use WBI to create a personal navigation history that can be accessed from every web page a user visits.

The disadvantages with the WBI architecture are mainly two. First of all, the plugins created by a designer has to conform to a specific language (in this case Java), making it less flexible than the SN. Secondly, for a user to gain access to the WBI it has to be downloaded and installed on the user's computer. The SN on the other hand allows a designer to create a socially enhanced space that works with no additional client software such as a social web site that works with the user's regular browser.

3.9.2 METAWEB

MetaWeb (Trevor et al., 1974) is a toolkit for bringing synchronous CSCW support to the web. MetaWeb provides information about people currently visiting a web page and allows people to cooperate. MetaWeb's building blocks for cooperation are *users*, *locations*, and *sessions*. A user is the representation of a person interacting with MetaWeb. Users move between locations. A location can be anything from a representation of a web page, to some other type of location defined by the designer. Each location hosts a number of sessions. It is within a session that people cooperate, e.g. communicate synchronously. Sessions are not restricted to an individual location but can be hosted by several locations. Two users that are on different locations can, thus, cooperate in the same session given that their respective locations host the same session.

In MetaWeb each user object has a name and location allowing any client that is attached to the system to request information about users and their locations. Similarly, each location object always contains a list of sessions occurring at that location. Finally, the session objects by default contain a list of users currently in the session, a list of potential members that can join the session, and a description (defined by the designer) of the session. Any client that is attached to the MetaWeb system can always access this default information about users, sessions, and locations.

A designer who uses MetaWeb implements the end-user interface, and provides custom implementations of the user, location, and session objects. Thus, what MetaWeb provides is ways to find out where people are and what sessions are going on at specific locations. Other functions, such as chatting, have to be provided by the designer but are mediated through MetaWeb. A designer extends the functionality of the MetaWeb system by extending the basic objects that can be manipulated.

METAWEB: DESIGN PRINCIPLES

MetaWeb only supports real-time presence. Information about users and their locations are always provided by the system. It is up to a designer to implement additional ways of making users aware of each other. A simple way for a designer to extend the notion of awareness in MetaWeb is to provide location objects that record the people that have been there in the past.

MetaWeb has an access control mechanism that controls who is allowed to join a session and who is allowed to see other users. One way to impose trust is to set up sessions that only trusted users are allowed to join. Again, since the basic objects in MetaWeb (users, locations, and sessions) can be extended, it is possible to imagine more complicated ways to implement mechanisms for trust. The user object could, for example, be extended to include a flag referring to expertise in some area, as done in the SN.

Privacy in MetaWeb is implemented at two levels: the user level and the session level. In the same way as session access can be used as a means to implement trust, it can also be used to provide privacy for users. By controlling the people who are allowed to join a session, MetaWeb automatically provides privacy for the users in the session. At a personal level, access can be used to control users that are allowed to receive events from a particular user.

What are the different ways that MetaWeb affords social navigation? At the lowest level it only supports awareness of people, that is, the only information that is provided by MetaWeb are users and their locations. Other than that, it is up to a designer to define what sort of social navigation the end-user application should support.

MetaWeb uses the same approach as the Social Navigator and does not provide an end-user interface. The reason for this being that MetaWeb does not want to place any restrictions on the semantics of the end-user application, neither in terms of the actual user interface nor in how the user, location, and session objects should be handled.

METAWEB ARCHITECTURE

MetaWeb extends the web server capabilities by adding a second server with its own protocol and API for synchronous communication. The core functionality lies in the server and it is the designer's responsibility to set up a proper client interface.

At the heart of the MetaWeb toolkit is the server. It stores object, handles the access control mechanism, allows clients to register interest in events, and communicates events back and forth between clients. Events can be sent to sessions, to specific users, and to the system in general. Each event has four fields:

- Source – who sent the event.
- Target – where the event is going.
- Type – a descriptor defined by the designer.
- Content – the actual content of the event.

MetaWeb defines a set of system events that can be understood by every client. Other than that, it is the designer who defines application specific events such as sending text messages between users.

If a user joins a session she automatically registers interest in every event in that session. To this an application can specify what type of events that it is interested in receiving. For example, an application could register interest exclusively in general system events.

Since MetaWeb is not part of the actual web server it cannot be directly accessed from a web page, but has to be accessed from a Java applet (shown in Figure 3.11). This makes MetaWeb less flexible than the Social Navigator because the client has to be Java based. On the other hand, it supports true synchronous communication between users. The MetaWeb server can send events to a client at any point, i.e. the server does not have to wait for the client to request information, which is the case with a web server based approach.

3.9.3 SOCIAL NAVIGATOR VERSUS WEBPLACES AND METAWEB

In conclusion, the Social Navigator fills a gap between WebPlaces and MetaWeb in that it

- is language independent. Designers that use the Social Navigator are not restricted to use a specific programming language. Both MetaWeb and WebPlaces require designers to work in the Java programming language.
- has built in support for indirect and direct social navigation. WebPlaces and MetaWeb are focused on direct social navigation.
- can be accessed directly from a web browser. WebPlaces has to be downloaded to the client machine.

4 DESIGNING SOCIAL NAVIGATION SYSTEMS FOR THE FOOD RECIPE DOMAIN

With a general design approach for social navigation at our disposal it is time to try it out on a domain. The question we want to pose next is: "How can the Social Navigator and design principles be used to support social navigation in a real world scenario?".

As the title states, online food shopping is the target domain. The goal is to design a socially enhanced system that helps users navigate recipes. Online food shopping was chosen mainly because it seemed to be well suited for social navigation. In the following sections the reasons for this will be explained.

This chapter has four parts. First, the shopping domain is examined and the reasons for why social navigation would be useful in it are laid out. Secondly, the design of the system EFOL (European Food OnLine) is discussed. In what ways did we design for social navigation? Thirdly, the system is evaluated, both to verify that social navigation did work but also to verify that the toolkit can be used to build a socially enhanced system. Finally, we return to the drawing board and create another system named Kalas, (in Swedish meaning party), to show how a small qualitative evaluation can serve as an effective tool to inform design. In summary, the intent is to show how

- social navigation can produce different designs,
- the principles for social navigation can be applied in the real world,
- the Social Navigator can be used to implement social navigation,
- a small evaluation can inform the design process.

4.1 ONLINE FOOD SHOPPING

It is difficult to buy food online (Sjölinder et al., 2003). In a typical online grocery store, there will be approximately 10.000 different products to choose from. If people are unfamiliar with the store they can spend hours trying to find the food they want. Navigating such a space is not only time-consuming but can also be boring and tedious.

As shown by Sjölinder and colleagues (2000, 2003) some users will have more difficulties than others to efficiently make use of the existing online stores. In a study of an existing hypertext based online store, they show that elderly users spent in average twice as much time finding items on a shopping list than younger users did. In both age categories, users sometimes completely gave up when searching for certain items. In average, users spent 12 minutes to find 10 ingredients.

Product presentation, customer navigation and search for products should be major factors leading to acceptance or rejection of electronic shopping by consumers. Frostling-Henningsson (2000) showed that on-line food shoppers do not gain any time from buying food online, instead they appreciate flexibility in time and space. Shoppers feel that they can avoid the tedious, boring, food stores, but at the same time they loose the sensuous pleasures of seeing, touching and smelling the products. This is somewhat compensated by getting status among friends from being able to tell stories about how they buy food online. Taking the human aspects of shopping into account becomes especially important when we consider users who do not have the opportunity to go shopping regularly, e.g. people with low mobility. In a study by Richmond (1996) on shopping in a virtual reality environment, results showed that users also want to be able to access the social aspects of a physical store; they want to socialize with other people. In contrast, existing online food stores are all "dead" spaces where users fill out how many milk packages they want sent to their doorstep.

Online stores are viewed as tools, in the traditional computer science way, and not as places where people shop, meet friends, and socialize. Given the problems with navigation and the lack of social interaction and sensuous pleasures in the existing online grocery stores, the domain should be an excellent application for social navigation techniques.

Before going any further let us lay out our vision on how to enhance the online food store with social navigation. The trail we will purse is one where the food store is turned into a place in which users shop from recipes instead of groceries. The store is set up as a recommender system that constantly pushes new recipes to users by using collaborative filtering techniques. The idea can best be explained with a hypothetical scenario. When a user logs onto the food store, it will present a recommended recipe to the user. This recipe is the currently most downloaded for the category of users that the user belongs to. The user can add the recipe to her shopping basket, which in turn adds the ingredients from the recipe to the list of items that will be delivered to her home. The user can then ask for the next-best recipes that fit her category of users - much along the same lines as Amazon.com recommendations ("other people who bought this book also bought these books"). The recommended recipe will be chosen on the basis of three different characteristics that the user can manipulate: user groups (e.g. vegetarians), the category of food (e.g. Italian), and any particular ingredient that should be included (e.g. shrimps).

There are many problems that need to be solved in order to implement such a store. Many issues are not directly related to social navigation, e.g. how to connect the store to a physical one is one, how to turn lists of ingredients found in recipes to lists of groceries, or security issues concerning how payment is carried out. As these issues are not our main focus we will simply regard them as solvable, although we realize that some of them are very difficult to design and implement.

4.2 DESIGNING EFOL

A list of groceries (ingredients) does not necessarily tell us what to cook. Hence, it is difficult to recommend food based on what other users have been buying. We need to consider the different meals and courses that somebody cooks to be able to model users' food preferences. Thus, we decided to recommend recipes to users. The recipes we cook from convey a lot about our personality, which culture we belong to, and our habits.

EFOL (European Food On-Line) allows users to buy food by selecting a set of recipes whose ingredients are added to their shopping lists. Recipe selection allows for accumulation of user behavior so that we understand which groups of users are most likely to choose different recipes. Shopping from recipes also makes it easier for users to plan their meals ahead of time and buy all the ingredients needed without having to search for each one of them. Through adding pictures of the courses the shopping experience also becomes more appealing to the eye. Based on user clusters, the recipes are in turn grouped into recipe collections.

As the intent is to try various forms of social navigation, the store should also be a populated space – providing some form of direct social navigation. The recipe collections provide the natural meeting points. The idea is to make users aware of other users, their choice of recipes, and also to enable chatting to discuss recipes.

The interface can be found in Figure 4.1. In summary, the interface shows a recipe in the bottom-right window, and next to it the ranked list of recipes in the current recipe collection. Above the recipe a chat window for the recipe collection is opened. Finally, to the left, there is an overview map with all the recipe collections. Superimposed over each collection the currently logged on users are visualized with simple avatars. We now discuss the design in more detail.

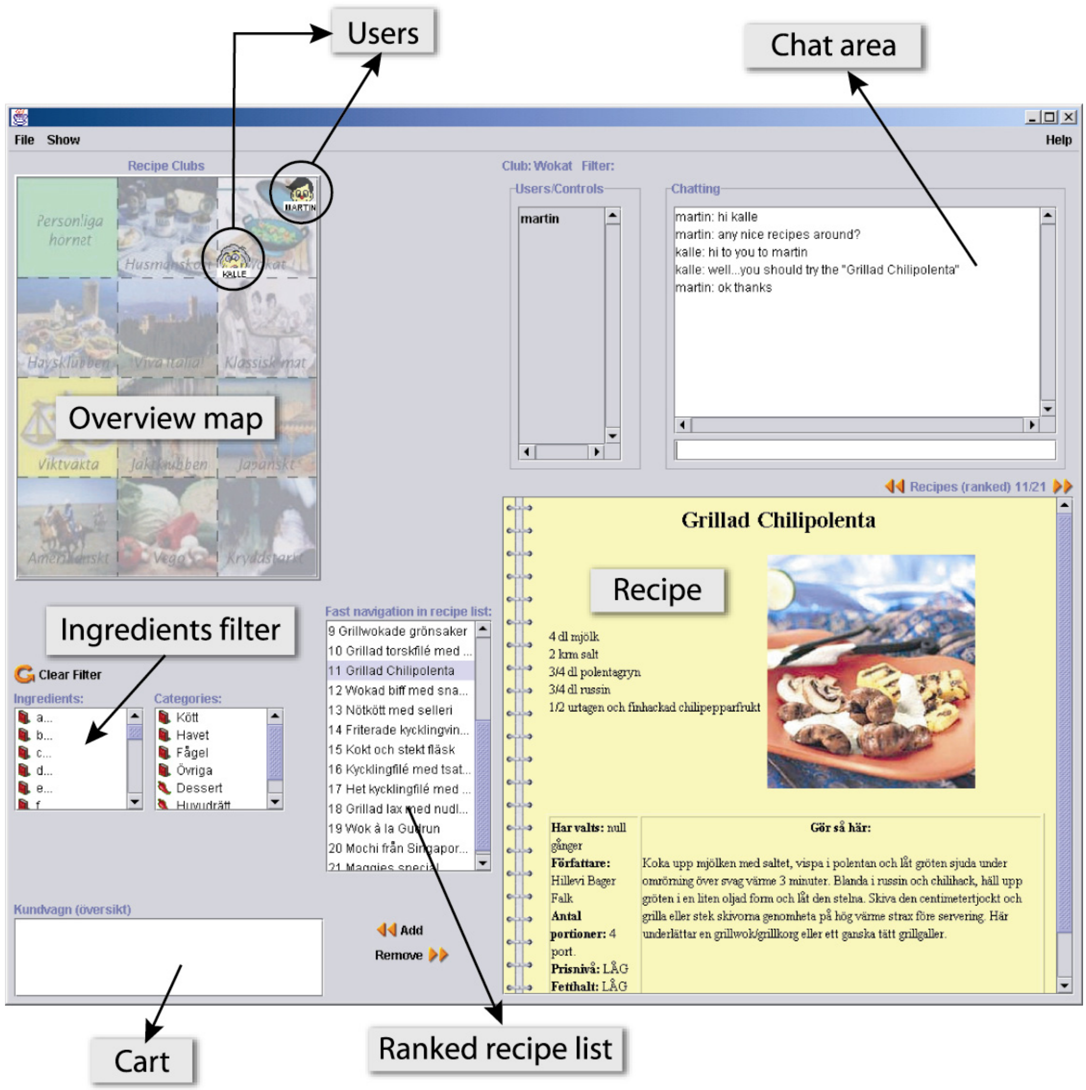

Figure 4.1: EFOL online store interface.

4.2.1 A DESIGN PRINCIPLES WALKTHROUGH

So far in the design process we have identified the basic structure of our online food store. The store should be based on recipes rather than individual groceries. We have also introduced the notion of recipe collections – places that have a special recipe theme. At the heart of the store lies a clustering module that recommends recipes to users based on individual preferences and what other people like, thus the store is a recommender system for recipes.

How do we proceed to enhance the store further with social navigation? The next thing to do is to investigate how the food store can be designed to incorporate more social navigation based on the Social Navigator (SN) framework and the following five design principles.

Presence: What are the possible ways to mediate presence in the online food store? Firstly, there is a relationship between recipe collections and users. Users move between them to find interesting recipes. In a way we can think of it as the online food store being the space and recipe collections represent different locations in it. An apparent way to mediate real-time awareness would be to map recipe collections to locations in the SN and present online users in different recipe collections in the online store.

Other important aspects of presence are trails of users. In the online store we identify three types of trails: (a) users' movements in the recipe collections, (b) the recipes should have information about how many times they have been chosen, and (c) the authors of the recipes should be visible in the recipes through their signatures.

Trust: At a first glance trust may seem unimportant in a domain like the online food store. But if we consider what type of recipe collections or users it can contain, trust will be important. A recipe collection can have an extremely specialized theme, say only gluten free recipes, in which it is important to trust other users' recommendations. In the same way it is possible to imagine that the store owner wants to be in the store to answer users' questions, and this needs to be signaled in some way to customers.

Privacy: Given that users in the store can see and talk to each other, we need mechanisms to ensure users' privacy. It could very well be the case that users do not want to be visible to others at all times. People should have the ability to turn invisible in the interface.

Integration: Social navigation should be integrated into the system it enhances. In the online store, recipe authors should be attached to recipes. The number of times a recipe has been chosen should also be integrated into a recipe. Finally, online awareness of users should be visual in the store.

Appropriateness: What type of social navigation is appropriate depends on the domain and task. The online food store is an experimental prototype so we want to include as much social navigation as possible and then investigate what works and what does not work. The store uses a variety of techniques for social navigation. It includes social filtering, various levels of presence, direct social navigation through chatting, and recipe read wear (how many times they have been chosen).

This brief walkthrough of the design principles illustrates a number of ways in which social navigation can be of use. The walkthrough demonstrated several potential problems that need to be considered in designing the store.

4.2.2 INDIRECT SOCIAL NAVIGATION IN EFOL

Recipes in the system are grouped into recipe collections. A collection is a set of recipes with a special theme, for example 'vegetarian food'. They are also places where customers can meet, socialize and get recommendations about recipes. Recipe collections are formed and given their names by their 'editors'.

Users can move around between collections to get different recommendations. Each collection contains a list of recipes that is ranked based on the usage pattern in that particular collection. In a way this can be viewed as the system giving personalized advice to users based on what others like.

While in traditional informational retrieval systems the search is based on the content in the existing items, recommender systems instead base their search on user behaviors (Resnick and Varian, 1997). One of the problems then becomes how to start, or bootstrap, such a system before any user behavior has been collected. This affects both the problem of items that have not been rated by users, as well as how to classify new users in the system. One solution is to combine a search based on a recipe's content with collaborative filtering. A user can thus constrain the recommendations given by the system, e.g., to find recipes that contain a particular ingredient. This gives users the desired control over the recommendations while it also gives the user a chance to find recipes that have not yet been rated by any user.

DYNAMIC RECIPE COLLECTIONS

A common problem with existing recommender systems such as GroupLens (Konstan et al., 1997) or Firefly (Shardanand and Maes, 1995) is that they give little or no feedback to a user on what user group she belongs to, or what user groups a recommendation is based on. The only feedback a user gets from the system is the recommend items, which is a poor way of reflecting a user's interests back to her. One problem is the rather complex task of automating the "labeling" of user groups. For instance, it would be extremely difficult for the Firefly system to automatically label a cluster of users as "reggae lovers with a flavor of ska".

Labeling of user groups not only tells the user something about which recipe collection she is at right now, but also gives information about other existing clusters of recipes and users. Labeled user groups enable a user to not only navigate the recommendations but also the user groups. In this way, a user can try out selecting recipes from different groups, thus getting access to a more diverse collection of recipes.

Our solution to the labeling problem is to put an editor back into the loop. There are two types of editors, the system editor and ordinary users. The system editor looks at log data collected from the recommender system and clusters users based on which recipes they have chosen and 'names' them with fuzzy names that convey somewhat of their content, for example, "vegetarians", "light food eaters", or "spice lovers". To enhance this process, the system is based on a filtering algorithm that uses an explicit representation of user preferences as recipe features (e.g. ingredients, fat level, time to cook) that change over time. It should thus be relatively easy for an editor to find similarities between users or recipes, and get an understanding of why they are similar. One could view the system editor as upholding the dynamicity of the system, i.e. over time users preferences will change, and so will the various recipe collections.

Ordinary users can at any time create a new recipe collection with a certain theme that she finds interesting, for instance, "Annika and her friends collection". These collections, obviously, do not have to reflect actual clusters of users as those found by the system editor. However, if a group of users choose recipes from such a collection on a regular basis their user profiles will converge. In this way users themselves are allowed to shape and model the space they inhabit.

Visual recipe collections might provide users with more insight into the social trails of their own actions as well as other users' actions that have lead to the recommendations they finally get. Recipe collections provide some clues to the inner workings of the recommender system. They could provide a basic model of the system's functionality that a user builds after having used the system for a while.

Real-time indirect social navigation

In addition to the recommender functionality, users have a real-time presence in the store through icons (avatars) representing them in an overview map of the recipe collections. As the user moves from one collection to another, the avatar will move in the overview map (see Figure 4.1). Our intention is to provide awareness of other online users. Since the user can see which collections are currently the most popular ones, this will hopefully also influence their choice of recipe collection and recipes.

Social annotations

We have argued that the social texture of information might be relevant. This is probably true for recipes: it is the style of the recipe, the author, the kind of life style conveyed by the recipe, or requirements on cooking skills, that matters when we choose whether to cook from the recipe or not. By adding the names of the authors and making it possible to click on them to get a description, users get a richer context for evaluating and choosing among the recipes. Each recipe also has a number denoting how many times it has been downloaded.

4.2.3 Direct social navigation in EFOL

The system also provides chat functionality tied to each recipe collection. Collections thus become 'places' in the information space. We could have chosen to make each recipe a place for chatting, but since the database contains over three thousand recipes, each recipe would not become sufficiently inhabited.

The implementation allowed users to be invisible, but in the user study we decided to disable this functionality, so that the effects of awareness and privacy issues could be studied.

4.3 IMPLEMENTATION

Some of the social navigation features presented in the previous sections were not implemented with the SN. Notably the recommender system was implemented with the framework developed in Laaksolahti (2002).

The key component in the online food store is the notion of recipe collections. Users can be in recipe collections, move between them, and choose recipes in them. When mapping *recipe collections* to *locations* in the SN, it becomes obvious how the toolkit can be used to implement the features just discussed. More formally, the SN allows us to:

- visualize recipe collections in an overview map,
- allow people to navigate between recipe collections,
- allow people to see other people in the system,
- provide chat functionality within recipe collections, and
- display navigational history in the form of user movements between collections.

PROVIDING THE OVERVIEW MAP

A major feature of the on-line store is the visibility of other users. A user can see who is on-line and where they are in the system. Recipe collections represent locations and each recipe collection is mapped to a location in the SN. The DefaultISN module (see Section 3.8) was used to present collections in an overview map, in which logged in users see their own as well as others' locations (or collections). At regular intervals (every 5 seconds) the overview map is updated to reflect user movements between collections. Figure 4.1 depicts the concept; there are 12 accessible recipe collections in this scenario, with two users currently in the system.

A reasonable question to ask is how to deal with more than 12 collections? The online store puts no restrictions on the number of recipe collections, but it is not feasible to use an overview map to display 100 collections. The solution to this problem is to delegate the responsibility of building the overview map to the user. The user herself selects the collections that should be represented in the overview map from a list of all recipe collections in the system (with no more than 12 collections visible at any time). Consequently, the map will not only serve as an overview of the space but also as a way of delimiting the space the user is in.

NAVIGATING BETWEEN COLLECTIONS

The DefaultISN module allows for navigating in space by clicking an overview map. This feature was used as the basis for navigating between collections in the recipe store. Clicking in one of the squares on the map takes the user to that particular collection. This is indicated by the user's icon moving from one square to another (this happens immediately) and that the list of recommended recipes is changed (see Figure 4.1).

CHATTING

In our on-line store we chose to implement chatting at a "community" level rather than at a personal level, i.e. no private chats are allowed. This configuration fits smoothly with the way the SN is implemented; since it creates a chat area for each location that it models. Users wanting to chat have to send and retrieve messages at the same location. This is what we mean by chatting on a "community level".

We used recipe collections to represent 'chat locations'. The idea was that only users in the same collection would be able to chat. In Figure 4.1 for example the users kalle and martin can chat. They are in the same collection (wok) and discussing if there are any nice recipes. Chatting is implemented with the DefaultDSN module. Whenever a user changes collection the module starts retrieving messages from the new collection once every second, to get a near synchronous communication channel.

VIEWING USER MOVEMENTS

Every time a user changes location, the SN stores this information as a key in the user profile. In the online store this information is retrieved and presented to the user. In this way a user can trace her previous movements between collections (i.e. she can follow her own footsteps).

Since the SN permits for other users to view the location log, this information becomes more interesting. Not only will it be possible for a user to follow her own actions (or movements between collections) but she can also look at other users' movements. Thus, if a user trusts another use's preferences for food she would want to see what other collections that user visits. An even more intriguing idea would be to allow people to actually see what recipes individual users have chosen in the past, giving them an even lower level of granularity in monitoring other users' actions. However, in such case, a designer has to be aware of privacy issues. Users have to be able to turn such monitoring off when they do not want to be logged.

4.4 A FIRST EVALUATION OF THE ONLINE STORE

In a qualitative user study we tried to establish to what extent our social navigation design intentions succeeded. We wanted to know whether the recipe collections aided users in filtering out good recipes, whether they were influenced by other users actions in the system (moving between recipe collections and chatting), and whether they understood that the system changes with its usage. We also wanted to check to what extent they experienced that their privacy was violated. In the study we hypothesize that:

- Social navigation has an impact of users acceptance of the system.
- Social navigation makes users experience with the system a more pleasurable one.

- Social navigation will affect the recipes users choose
- Some users will feel intimidated by social navigation

This small study served as valuable input in the following redesign of EFOL into Kalas.

4.4.1 SUBJECTS

There were 12 subjects, 5 females and 7 males, between 23 and 30 years old, average 24.5. They were students from computer linguistics and computer science programmes. The two groups did not know one another before the study. None of the subjects had any experience of online food shopping prior to the study.

4.4.2 TASK AND PROCEDURE

The subjects used the system on two different occasions. They were asked to choose five recipes each time. Their actions were logged, and we provided them with a questionnaire on age, gender, education, and a set of open-ended questions on the functionality of the system. They were given a food check of 300 SEK (~$30) and encouraged to buy the food needed for the recipes. (In the current implementation, the recipe recommendation is not yet connected to any existing online grocery store that can deliver to the home.)

4.4.3 RESULTS

Overall, subjects made use of several of the social navigation indicators. They chatted (in average 6.5 statements per user during the second session), they also looked at which recipe collections other users visited, and followed them around. About half felt very influenced by what others did in the system.

Concerning the effects of adding pictures we found that half of the subjects got hungrier from using the system than they were before starting. 75% of the subjects were in the same or a better mood after using the system (despite a set of technical breakdowns during the sessions).

4.4.4 DIFFERENT USERS DIFFERENT CONCERNS

After using the system subjects answered the question “Do you think that it adds anything to see what others do in this kind of system? What in such a case? If not, what bothers you?” One subject said: “I think it is positive. One can see what others are doing and the chat functionality makes it more social. One could get new ideas”. Not everyone was as positive: “No! I cannot see the point of it, I have never been interested in chat-functions”.

We looked closer at this difference and found that the subjects could be divided into two groups. One group, consisting of 10 subjects, who felt influenced by others, and one minority group, consisting 2 subjects who claimed not to be. The logs from their sessions with the system also backed up this difference: the first group chatted and moved between collections without hesitation. In their comments, they also stated that visible activity in recipe collections influenced them: they were attracted to collections where there were other users and they became curious about what the other users were doing in those collections. When asked about other services they would like, they were positive towards functions such as sharing recipes with a friend, more contact with the owner of the food store, and being able to comment on a recipe and see others comments.

The remaining two subjects were consistently negative towards social trails. They did not like to chat, they disliked being logged, they did not want more social functions added to the system, and they could not see an added value in being able to see other users in the system. Their claims were again backed up by log data: they did in fact not chat, and one subject did not even move between recipe collections.

The division into the two groups is also in line with whether they would like to use the system again. The ones who did not want to use it were the ones who claimed not to be influenced by the actions of others. Interestingly enough, more or less all participants found the system fun to use even the ones claiming they did not want to use it again.

When investigating subjects' answers to the open-ended questions, certain aspects of social trails in the interface do not seem intrusive at all, while others are more problematic to some users. The fact that the recipes show how many times they have been downloaded is not a problem – it is not even mentioned. Neither is the fact that choosing a recipe will affect the recommender system. When asked whether they were bothered about being logged, one subject answered: "It does not bother me at all. There are so many facts about me spread everywhere anyway, so what does it matter? Besides, one gets logged in order for the recipe recommendations to get more individualized and that should lead to saving time." This view keeps coming back: as long as there is a perceived benefit, and the name of the user can be faked, most users do not mind being logged.

The two users who disliked being logged answered: "Well, maybe. I do not like being logged" and "It does bother me somewhat that others can see what I do, for example I did not jump as much to the other recipes collections but stayed in the 'Personal Corner' [the default collection] because of this. But when it concerns something like a recipe I do not think that it matters that much whether I get logged or not. One is relatively anonymous anyway since one does not log in with email address or any other personal information." Thus, seeing the avatar moving between recipe collections is more intrusive than the fact that choosing a recipe affects the recommender system.

Most subjects felt influenced by how the other users avatars moved between recipe collections: "If many stand in a collection one gets curious and wants to go there and check it out" or as another subject said: "I got somewhat distracted from seeing them jump around. It was a little bit exciting when someone else entered the same collection [as me]". A worry we had as designers of the system was that users would feel that they were not rewarded when following other users. Once a user has moved to a collection with many visitors, the only thing that changes is that she can chat with them. She cannot see which recipes they are looking at nor which ones they have chosen from this collection so far. The list of recipes in the collection are of course ordered by how popular they are, but this ordering is not only based on the actions of the concurrent users, but is also inferred from what other users, not currently logged into the system, have chosen in the past.

Chatting is even more intrusive than the logging and avatar movements. As one user pointed out, he did not mind getting his choice of recipe logged, but if the contents of the chatting would be used and saved in the system, he would be bothered. But again, chatting was only intrusive to some. Most (9 subjects) saw it as a positive addition.

In general, being logged does not bother users – they know that this happens all the time anyway, and they do not mind sharing their preferences for food. It is when their actions are not anonymous and other users can 'see them' that a minority of users react negatively.

4.4.5 Social affordance

Through adding social trails to the interface we hoped that it would encourage users to explore the space, and perhaps guide them to a better understanding of the functionality, "the appropriate behavior", in the space. One subject said: "Yes, I found it interesting to be able to see what the others were doing. The only thing that bothers [me] is if one sees them doing something and then one does not understand how to do the same thing. Right at the beginning I did not understand how to switch recipe collection and then it was frustrating to see the others change collection all the time". While this subject expresses frustration, she still captures our intention: to reveal system functionality through making other users' actions somewhat visible.

4.4.6 Social experience

Adding social navigation to the design definitely made our subjects feel that it was a social experience: "The system became alive and more fun when one could see the other users". Another user said: "I think it is good to introduce social contact. In many systems on the net, several users may be logged in, but you cannot feel their presence". One user said that the best part of the system was "To have a chat function so that one does not feel all alone in one's struggle against the system." Users also asked for other forms of social functionality, such as being able to share a recipe with a friend, being able to chat with the owners of the store, getting in touch with professional chefs, or publishing a week menu for others to be inspired by.

4.4.7 Understanding recommender functionality

Despite the fact that these were students from a course on intelligent user interfaces, they did not have any clear picture of how the recommendations happened, or why there were recipe collections. They hypothesized that the order of the recipes was affected by their choices, but they did not have any clear theories of why or how this happened or how it related to the recipe collections. Even if our subjects did not fully understand the dynamicity of the recipe collections or the fact that they represented user groups, they got a better insight into the workings of the system than would have been possible if there had been no recipe collections at all.

One user said: "Yes, maybe a recipe that often gets chosen is more representative of the recipe collection and can be recommended to others".

4.5 Moving from EFOL to Kalas

In the previous sections we have seen how the Social Navigator can be connected to a system to enhance it with social navigation. Indirect and direct social navigation were added to an online food store. The design was concluded with a small user study that showed some trends on how to design a social navigation system.

The new system that will be developed is called Kalas. It is in part based on what was found in the EFOL study. It will also be designed to work in a real-world setting. The section begins by discussing the design recommendations found in the EFOL study. Since the underlying architecture of EFOL and Kalas are the same we will not go through how Kalas is connected to the Social Navigator.). The recommender system was however replaced with the Predictor framework that was touched upon briefly in the previous chapter (Cöster, 2002).

4.5.1 Design considerations from EFOL

While EFOL was a small-scale study that needed to be redone in a more realistic setting with a larger number of users, it pointed at some strengths and weaknesses in social navigation that need to be further investigated. Strengths of the EFOL solution were that it was indeed a social, pleasurable and entertaining system. It succeeded in turning the space of recipes and ingredients into a *place* with recipe collections.

Weaknesses that need to be considered when designing for social navigation includes first of all ensuring a stronger privacy protection for those users who wish to be anonymous. Two out 12 users who evaluated EFOL felt that their privacy was invaded when others could see them move in the EFOL overview map. In a real system the possibility to become invisible needs to be included. This can easily be achieved with the Social Navigator toolkit (see Section 3.5.3) by adding a flag to each user signaling her invisibly.

Kalas has to enable users to give more elaborate social input to the system. In EFOL users lacked the ability to comment on individual recipes. They lacked the ability to give explicit feedback (both negative and positive) on previously selected recipes. Comments and votes account for two important ways of expressing opinions about recipes. Explicit feedback (or votes) could, to some extent, reduce the potential snowball effects found in a system that relies heavily on implicit feedback from its users.

A problem with EFOL was that the overview map could potentially be misleading. Users in it were anonymous and there was no way of telling them apart. The interface did not allow users to quickly get an overview of who was logged in and what their purpose of being in the system was. Through creating different visualizations of users, friends, chefs, and storeowners users could easily choose whom to follow, rather than just blindly follow an anonymous crowd. Allowing users to make more informed choices of whom to follow should reduce the snowball effects arising from just "following the crowd".

Another problem that became apparent in the study was how to convey the recommender system functionality. Users had no clear idea on how recommendations were calculated and only vague notions of them being a product of other users actions. One user said: "Yes, maybe a recipe that often gets chosen is more representative of the recipe collection and can be recommended to others". A possible solution to this problem would be to show the contents of the users profile and how that affected the recommendation. Another way would be to show the contents of the user profiles of those people that affected a recommendation. These two techniques could of course be combined and even be made interactive, thus allowing users to experiment with their profiles to see what effects it would have on recommendations.

As a final observation we noted that the terminology used in EFOL caused some confusion. Recipe collections were called clubs, which implied membership. You are a member of a club and it is closed to outsiders. In EFOL users felt they belonged to certain clubs, which was not our intention. This may seem a minor issue, but if it affects users' navigational behavior it becomes problematic. The simple solution was to rename clubs to collections to avoid any confusion.

4.6 Kalas

From Figure 4.1 and Figure 4.2 we see that EFOL and Kalas look completely different. More or less everything at the interface level has changed. The basic features are still there. There are recipes and an overview map with the visible users. Users can chat and edit lists of ingredients. To a large extent Kalas was designed to address the issues that were unfolded in the EFOL study. The one issue that was not covered was how to convey the recommender system functionality. The reason for this was purely pragmatic; moving away from the "black box" approach of a recommender system to more of a "glass box" approach requires extensive research which was not the focal point of this thesis. Instead only small changes were made to clarify the recommender system.

Kalas is not only a redesigned version of EFOL; we also wanted a new and more appealing look. Furthermore, EFOL worked under the presumption that it would be connected to a physical grocery store, which is not the case with Kalas. This requirement alone forced us to rethink the incitement for using the system. More functionality had to be added to make the system worth using and some of the terminology used in the first system was changed. For example, the shopping cart in EFOL was replaced with a shopping list in Kalas.

4.6.1 Overview map

In both EFOL and Kalas people navigate the system by moving between recipe collections in the overview map. In EFOL the overview map consisted of collections grouped in four rows and three columns. The problem with that configuration was that users did not clearly see the boundaries between collections. Another problem is that ordering the space this way induces a relationship between collections. We create both a semantic and physical structure (see Section 2.1.2). If there are no relationships between collections we want to minimize any notion of such.

In Kalas the map, consequently, was reconfigured to fit with the more unordered list of recipe collections. The overview map is flat and collections are clearly separated to minimize the confusion of where users are. The way users navigate between collections was also changed. Rather than clicking in the map itself (as was done in EFOL) users click a button next to each collection. There are two reasons for this change: users are accustomed to clicking buttons, and collections can be very crowded without users not finding any clear spot to click on.

The overview map in EFOL was static and only allowed for 12 (4 * 3) recipe collections all together. In a real scenario it is preferable to not limit the number of collections in this way; it should be up to the users how many they would like to use. As was pointed out earlier, collections could potentially be created by users, editors or even generated automatically. The overview map in Kalas therefore supports scrolling of collections. Should there be more collections than can be displayed users can scroll the map. This can be achieved with a map built up by rows and columns. However, introducing more dimensions (i.e. left and right navigation) could potentially make the map more difficult to navigate.

The last point that should be mentioned is that the avatars have been removed from the overview map (see Figure 4.2). Some people in the EFOL study felt that the avatars were disturbing, especially when they only had a small selection of avatars to choose from. An avatar is a reflection of a user, thus, users need more control over them. Removing the avatars also allowed us to visualize more users in each collection. Displaying a text string requires much less space than displaying a picture.

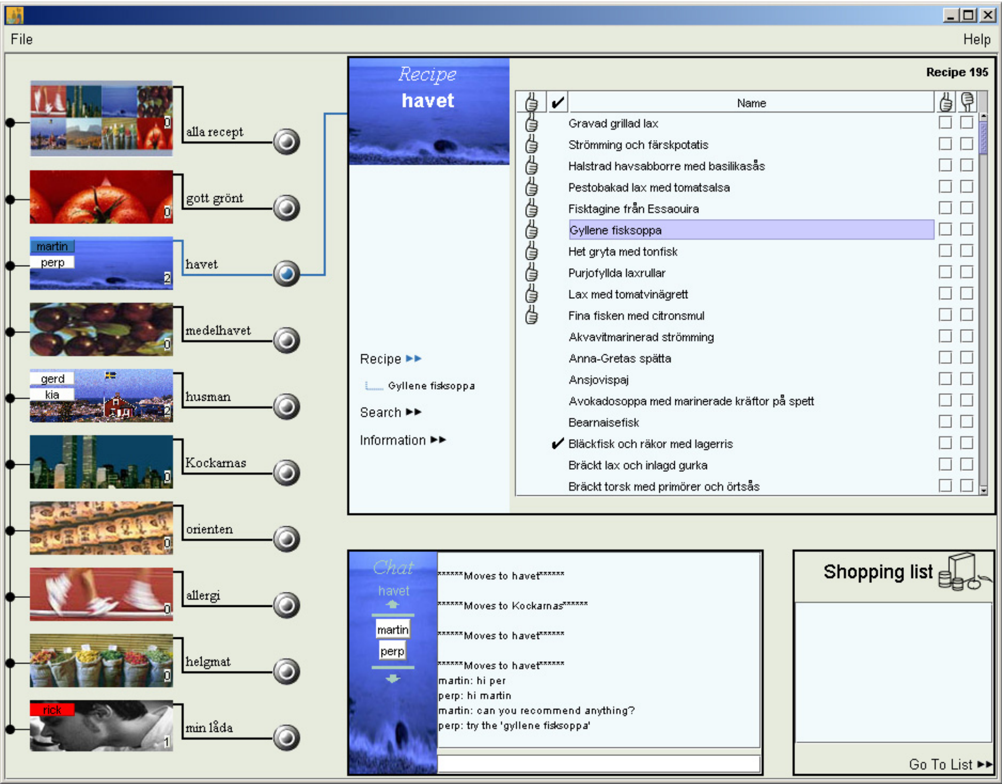

Figure 4.2: Kalas recommended recipes.

Figure 4.3: Kalas recipe.

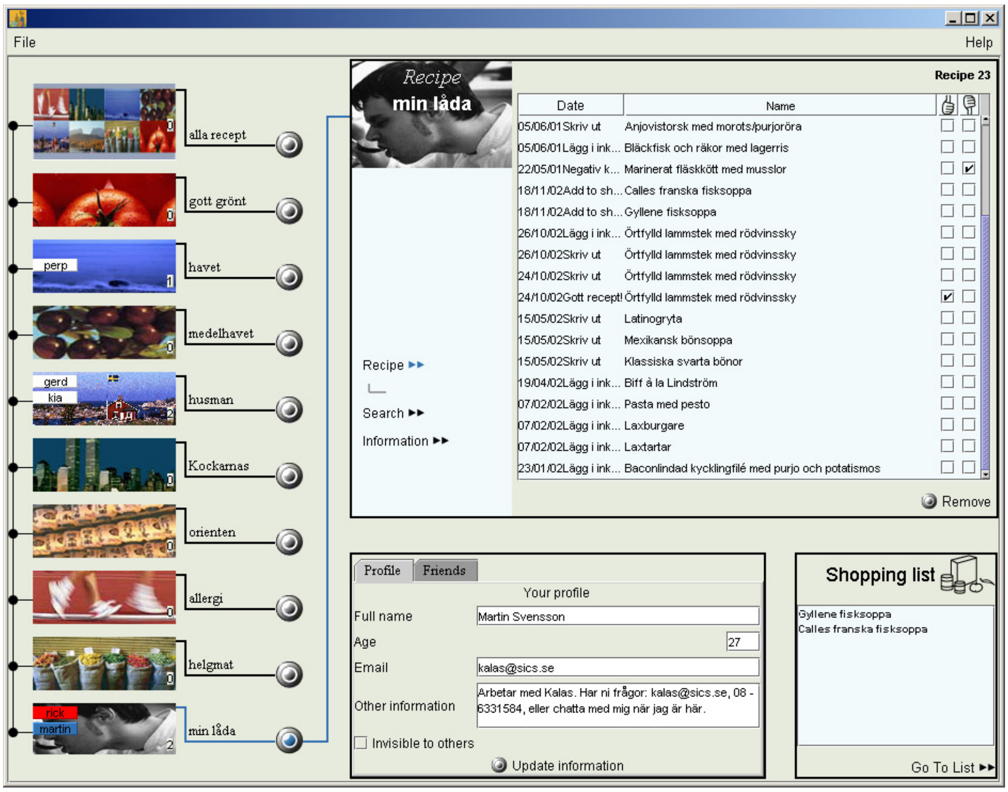

Figure 4.4: Kalas "My Box" view.

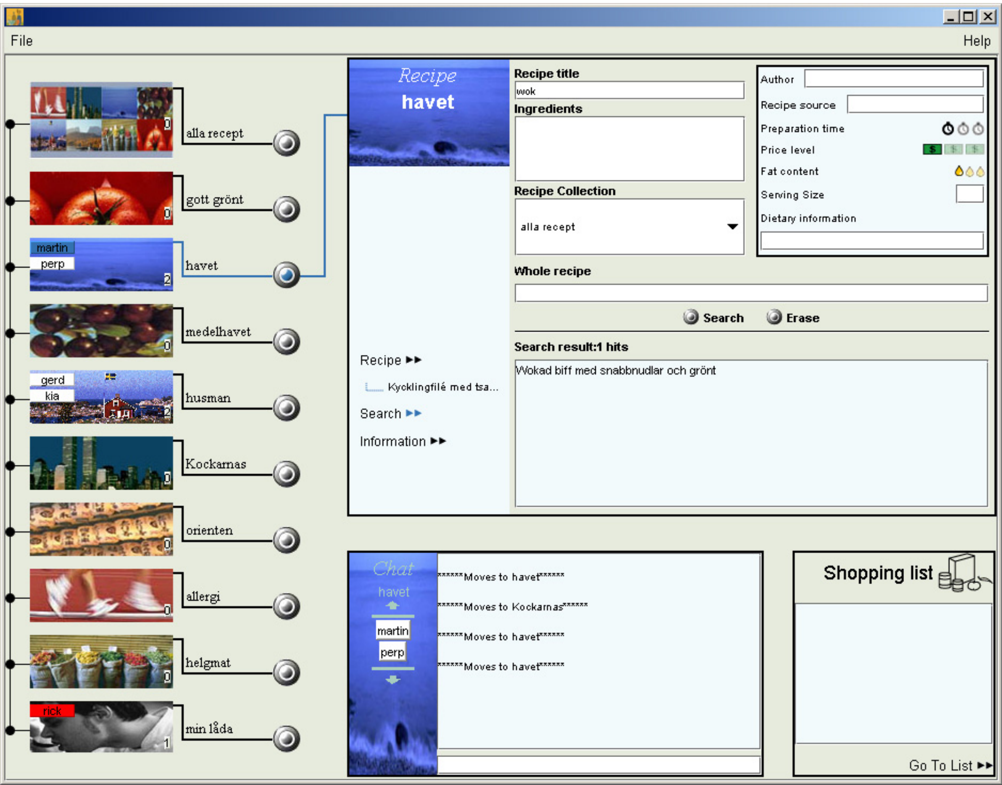

Figure 4.5: Kalas search view.

4.6.2 RECIPES

Individual recipes have undergone quite a few changes (see Figure 4.3). Starting with the physical position, recipes in Kalas are displayed in the top-right corner, giving them a more central place in the system. The social information attached to each recipe (author, source, number of times chosen, and so on) is made more visible by placing it at the top of a recipe.

As was suggested in the study, recipes can now be commented on. An individual recipe can contain several comments, enabling simple discussion threads around each information item in Kalas (i.e. recipes)[12]. Users have the ability to give explicit feedback to recipes by clicking the "thumbs up" button. Comments and feedback makes the recipes in Kalas more interactive. They give users the opportunity to place social trails on recipes for others to see.

In EFOL the idea was to create a system in which all information was visible at any time. Users could see the ranked recipe list, the overview map, and individual recipes all at the same time. The downside was that everything had to be small to fit the screen. In Kalas – with more interactivity added – we decided to loosen up this requirement. Browsing and viewing recipes are separated and carried out in different system modes in Kalas (see Figure 4.2 and Figure 4.3).

4.6.3 EXPLICIT AND IMPLICIT RATINGS

The recipe changes and overview map changes are to some extent cosmetic. There are other changes that to a greater extent affect the underlying system in Kalas, namely the way users are allowed to interact with recipes. It will not only make recipes look different but also affect how recipes are recommended to users.

In EFOL, which essentially was an online grocery store, we argued that implicit feedback was enough as input to the recommender system. The argument was that sending a set of groceries derived from a number of chosen recipes was a very strong positive action on those recipes. Users are very careful before actually spending any money. In Kalas, however, we removed this connection. We turned the system in to a plain recipe navigator. The main reason for doing so was that it was tested on real users. To keep the notion of a recipe store would have forced us to implement a live connection to a physical grocery shop.

What Kalas offers is not to send shopping lists to a store but to print them and take them to a store. This entails two radically different actions in terms of "risk" on the users part. For the recommender system the change could potentially lead to more noise in user profiles. That is, implicit positive votes that do not accurately reflect a user's taste. Thus, we had to find additional means for giving feedback to the recommender system. The implicit feedback mechanism has a great advantage over explicit feedback, it allows for rapid collection of user feedback without putting any extra burden on users. To solve the problem we

[12] An interesting extension to comments would be to implement anchored conversations (Churchill et al., 2000), allowing users to start discussions anywhere in the system.

- added a special collection ("my collection") that can be used to store recipes.
- made it possible to get print outs of recipes.

In effect, Kalas not only offers one way of giving implicit positive feedback but three: printing recipe lists, saving a recipe in a special collection, and printing individual recipes. Still, the level of noise can potentially be high, which indicates that some explicit feedback mechanism is necessary. Also, the EFOL evaluation indicated that users wanted to explicitly state their opinions about recipes. Explicit feedback was thus added. To keep things simple for users we chose a binary scale: "thumbs up" and "thumbs down". Explicit feedback has precedence over implicit feedback, meaning that, for example, clicking thumbs down will always render a negative value on a recipe no matter how strong the implicit positive feedback on that recipe was. It should also be mentioned that the recommender system does not accumulate repeated feedback on a specific recipe.

4.6.4 My Box

My Box is the special recipe collection where users store recipes, change their profiles, go invisible and mark other users as their friends. The idea with My Box is thus to add more value to Kalas. To start with, all actions that are picked up by the recommender system feeds (i.e. recommend, print, save, etc.) are visible in the collection. Instead of having a recipe list with a special theme (c.f. the other collections) My Box contains all recipes that a user has previously chosen and what action that led to it being stored in the collection. Each recipe is sorted by date, meaning that the most recently chosen recipe by default is on top of the list. In Figure 4.4 we see that the last action the user did was to print the recipe "Ansjovistorsk med morots/purjoröra". Thus, My Box acts as a recipe diary if you will and a place to mange user profiles.

How can the recipe history be used? When a user enters the system she automatically starts in My Box. This means that she immediately sees what she did the last time she visited Kalas, and more importantly, gets a chance to react on those recipes without much effort. In that way, the collection offers an easy way for uses to give explicit feedback on previously chosen recipes, which is of course important when using implicit feedback as the main instrument for building user profiles. Since the explicit feedback overwrites implicit feedback, the feedback users give in My Box enables the recommender system to correct errors in the user profile. For example, if a user prints a recipe (a positive action) and then later comes back and votes negatively on that recipe the value in the user profile will be update to reflect a negative opinion towards that recipe.

There is no chatting facility in My Box, which should come naturally if we consider the collection to be the users' private corner. Also, the recipe lists in My Box is different for every user (e.g. the recipe history), which makes it difficult to discuss the recipes in that collection. Not all social features are lost when moving to My Box. Users are still visible in the overview map and others can view their personal profiles. Making My Box completely private (i.e. users do not turn up in the overview map) could confuse other users. Users would not be able to tell the difference from being online, in My Box, or invisible. Then again, those users who whish to be invisible in My Box can actively turn invisible.

4.6.5 The ranked recipe list

EFOL used an extremely simple recipe list with recommendations displayed in descending order. All recipes in a collection were ranked according to the recommender system and no other way to sort the list was possible. The only way a user could influence the recipe list was with content-based filters. It was pointed out in Section 4.4.7 that users did not clearly understand the recommendations and did not make the connection to other users' actions. As a solution we primarily discussed ways of visualizing user profiles to explaining the recommendations. Because of the complex nature of such a solution it was decided to try a much simpler solution. Instead of ranking the whole list of recipes by recommendation, only the top 10 ranked recipes would be sorted by recommendation and the remaining ones sorted by title. To further visualize the recommended recipes we tagged them with the "thumbs up" symbol. By using the same symbol for recommendations and explicit positive feedback we hope that the relationship between voting on recipes and getting recommendations should be clearer. Although this is not an explanation per se, the recommendations ought to be more visible to users.

4.6.6 Content based searching

In real life a user often has a particular goal in mind. In terms of cooking it could be to make a nice dinner for your family or make something nice with the "left-overs" you have at home. Social navigation, and recommender systems in particular, aid users in the more exploratory task of finding new and exiting recipes. However, if a user knows she wants to cook something with fish in it, it could be cumbersome to browse through several recommendations before finding a recipe containing fish. Even worse, the recommendations may not at all contain the ingredients the user needs. In EFOL the problem was solved by allowing users to specify filters based on content that only showed the recipes that matched the filters (see Figure 4.1).

The filter approach turned out to be difficult for some of the students in the first evaluation. They did not see the active filter (it was shown just about the chat window) and did not realize that the filer was still active when they moved between collections. This resulted in some empty collections because none of the recipes in them fulfilled the criteria set up. This is a typical interface design flaw, which could be solved with a better design. Nevertheless, it is probably sensible to separate searching from filtering. They have distinct purposes and should therefore be kept apart.

Figure 4.5 shows the search facility in Kalas. Instead of using the filter metaphor just explained, Kalas uses an ordinary VSM[13] free-text search, with results displayed in its own recipe list. One interesting extension would be to allow users to run the

[13] The Vector Space Model (VSM) is the dominant model for free-text searching (Baeza-Yates and Ribeiro-Neto, 1999).

results through the recommender system to get the list ordered by recommendation rather than by content. This is an example of a more natural way of combining content based searching with collaborative filtering.

The most interesting aspect of the search facility is the metaphor employed. It tries to mimic a recipe as much as possible, making it easier for users to search for recipes by author, ingredients, cooking time, and so on. By default, a recipe search is restricted to the current collection a user is in, thus keeping the notion of being in a specific recipe collection.

4.6.7 INVISIBILITY

Privacy issues are important in these kinds of systems. We know that there is a group of users – although small – that are less inclined to share social information with others. Social navigation relies on peoples' willingness to, in various ways, share information about themselves. Shall users who choose not to share benefit in the same way as those who do? Our view is that the more a user contributes the more she should gain. Kalas hence gives users the ability to be invisible at the price of loosing some of the "social functionality" in the system. Invisible users can give feedback on recipes, comment on recipes, see other users in the overview map, but they cannot participate in chatting, both in terms of writing messages and reading messages. It is not difficult to motivate such a design choice. To give two reasons:

- To participate in a discussion when no one knows you are there is not fair to those who believe they are the only people listening Observe that this does not mean that lurking (Noonecke and Preece, 2000) is not allowed. Lurkers are known but not active participants.
- Social navigation builds upon people exposing themselves. It is natural in such settings that some features would be lost if a person chooses not to expose herself. In a sense users only gain as much as they contribute.

In Kalas we chose only two visibility modes, on or off. The control of invisibility could be more fine-grained than that, for example, users could be "not up for chatting" or "busy at the moment". Various degrees of invisibility could allow more or less access to the social functionality in the system.

The transition from EFOL to Kalas is noticeable. Most of the interface has been redesigned and some additional functionally added. When designing a system for a real-world scenario there are issues that need to be dealt with. We have to make the system appealing to users. What functionality it provides and how it looks are factors that are not nearly as important in a prototype studied under lab conditions. Another issue that was not mentioned here is stability. Making the system stand on its own requires a considerable effort from the system designer. We do not claim Kalas to be the ultimate recipe recommender system, although it captures many of the aspects of social navigation that we want to study. Some design considerations that were unfolded in the pre-study were not addressed – notably little effort has been spent on addressing how to best explain recommendations. In the following two chapters we take a closer look on how the Kalas system was evaluated.

5 EVALUATING SOCIAL NAVIGATION

In many domains and situations there have to be a certain number of people available to measure the effects of social navigation. In a small study it is often only possible to observe trends and possible effects of a phenomenon based on social behavior. The main benefit of such studies is to inform on design. The third research challenge proposed in this thesis is yet to be tackled. We still have little knowledge on how and why social navigation is of benefit to a navigator in real life.

The remainder of the thesis will be devoted to this final question. This chapter focuses on how to evaluate social navigation. What are the questions that need to be answered to help us decide if social navigation works or does not work? What are the right settings for studying it: should it be done in a controlled lab environment or totally unsupervised in the "real world"? A number of questions (or evaluation criteria) are derived and later used to evaluate the Kalas system discussed in the previous chapter.

It is not possible to conclude that social navigation will work in general from one or two studies, or for that matter, arrive at a conclusion that it does not work at all. But what can be done, is to place one piece in the puzzle of attempting to understand how to move from seeing how social navigation happens in the world, to seeing how we can design for it in the virtual world – be it tied to one particular domain and group of users. To lay this piece in the puzzle, we first have to tease apart the reasons as to why social navigation could be of benefit to users and what the potential drawbacks are?

5.1 THE BENEFITS AND PROBLEMS WITH SOCIAL NAVIGATION

In the previous chapter we saw that the social navigator toolkit can be used to implement a system with social navigation features. It is now time to address the actual evaluation of those social navigation features and the specific questions that need to be answered to see whether they are beneficial or not.

In Section 3.2, five design principles for social navigation were put forth: awareness, trust, privacy, appropriateness and integration. These are principles that have to be considered when designing for social navigation. They do not necessarily say anything about how the effects of social navigation will manifest themselves. For example, when evaluating in what ways social navigation helps a user find her way in space it is not enough to conclude that the system supports real time awareness of other users. One can view it as the design principles inform us on how to design the system and the evaluation criteria are tools that help us measure the effect of the social features in the system. There is, however, an inherent relationship between the design principles and evaluation. An evaluation could, for instance, show that real time awareness does not work, in which case that particular principle has to be reconsidered. The design principles are rooted in the way social navigation happens in the real world, which does not necessarily transform to the type of computerized system we want to enhance. It is important to realize this and set up the evaluation criteria in a way that it is possible to conclude that, for example, online awareness is not important for social navigation in electronic systems.

In the beginning of this thesis we claimed that there are a number of ways in which social navigation is of benefit to users. Although small, the EFOL evaluation supported our claims on this account to some extent. The results indicated that social navigation did in fact help users filter information and allowed the online food-shopping domain to be turned into a social place. What is now needed is a deeper understanding of how it helps users. What does it mean when we say that social navigation helps users *filter* out information? Why is the *social texture* and quality relevant to a user? Will *social affordance* happen and how? What are the implications of users *changing the layout of the space*? Only when we have a deeper understanding of these questions is it possible to evaluate them.

The benefits with social navigation are also what make it difficult: namely that it requires a group of people to take place. This inherent requirement forces us to evaluate social navigation both in terms of its benefits and its potential problems. The requirement will render a set of problems, or design opportunities for that matter, that should be considered when evaluating social navigation. If we agree that "the more people that are available the better" social navigation is expected to work poorly in a system with few people in it, i.e. social navigation faces a *bootstrapping* problem. By this we mean that the ratio between the number of people and the size of a space must be sufficiently large. It is not necessary that any given piece of information needs to, for example, be commented by many users. Instead, if there are many pieces of information in a space a sufficient number of users have to be available to leave their social trails. Another problem is that social navigation can create *snowball effects* when more and more people follow the wrong path.

The EFOL study demonstrated that the level of *privacy* support needed varies among users. When designing a system that supports social navigation privacy has to be considered. A good design should allow users to protect their privacy.

People and information change over time. People's preferences change so that they no longer have exactly the same needs, attitudes, or taste. The actual information in the space might also change. In an information space of research reports, with new reports constantly entering the system, only some of the earlier reports will be relevant over time. Thus for social navigation to work the system will have to deal with an issue that we shall name *concept drift*[14]: that is, how the meaning of concepts or preferences of users' drift over time.

[14] Concept drift is frequently used in the Machine Learning community as a term for a concept's change in meaning over time (Widmer and Kubat, 1996).

Adding to these issues, it is also desirable to look at pure design aspects of social navigation. In what ways can systems be designed, in terms of icons or numbers or ways of displaying information paths, to get the most out of social navigation?

The following sections discuss each potential benefit and problem in greater detail. The aim is to arrive at a series of key issues that a designer can consider, or use as a starting point, when evaluating a system that supports social navigation. Since our work focuses on the recipe domain the following discussion will be conceptualized with examples from that domain. Based on the arguments outlined here, we show in Chapter 6 how Kalas was evaluated with 302 users over a six-month period.

5.2 FILTERING

In many situations in the real world people will just follow and do what others have done in the past. A prerequisite for this to work is that people see the social trails that surround information and roughly understand their meaning. Once people have noticed and understood the social trials (in some sense) they should feel (subjectively) influenced by them. Observe that this works in both directions, either a user can have positive feelings towards the social trails ("I want to choose what others have recommended and chosen before me") or negative feelings ("I do not want to choose this since others have chosen it before me"). In either case the social trails are of benefit to a person and signs of success of the social filtering mechanism.

The problem is, of course, what does it means to filter out interesting information? Given the reasoning above, if users are subjectively positive towards a recommender system and use it to find some of their information – it works. If we argue along these lines the question is easily answered, we simply ask the users for their subjective views, e.g., "Did you find the recommender system useful?". However, in any information system we argue that it is necessary to tease apart the reasons for choosing a particular piece of information. There are a number of decisions a user makes throughout the process of finding the right piece of information. In many situations what makes some information preferred over other information is not at all obvious. In terms of food recipes a user makes a number of conscious and unconscious decisions before finding a good recipe. It could be the recipe's ingredients, the way it looks (i.e. a photograph of the dish), or perhaps its popularity that triggers a user to choose a particular recipe. Whatever it is that triggers a user to choose some particular information, it is a complex process and most certainly dependent on several factors. A couple of scenarios where a user chooses some information will illustrate the usefulness of a recommender system.

The information was useful and also recommended. In this scenario we do not know if it was the recommendation that triggered a user to choose the recipe. However, the recommender system did its job and recommended something useful. Thus, in this case, it is safe to conclude that the recommender has an accurate picture of what is interesting to a user, albeit the reason for choosing the information is hidden from us.

The information was useful but not recommended. In this scenario we can determine that it was not the recommendation that triggered the action (which is what is interesting to us), and hence, conclude that the recommender system failed. The reasons for failing, however, can be many. It could be that no one else has seen or chosen the information, which would mean that it was a *bootstrapping* problem that caused the system to malfunction. Another possibility is that the system has an inaccurate model of the user in which case it has to be updated.

The information was not useful but recommended. In this scenario a user first chooses a piece of information (based on a recommendation), but then, for some reason, does not find it useful. Thus, the user can either choose to act on the recommendations or not. In any event the system would need feedback from the user to be updated to better fit the user's needs and preferences. This situation can potentially lead to the type of *snowball* effect that was mentioned earlier.

The information was not useful and not recommended. As in the first scenario, this is a situation in which the recommender system has an accurate model of the user, at least to the extent that it does not recommend the wrong information. The problem in this scenario is of course, that if a user keeps on choosing information that is not useful the recommender system will end up having an inaccurate model of the user.

A lot of information is useful out of which some is recommended. In a normal situation one would expect not only one but many information pieces to be useful (c.f. food recipes). Arguably, this is where a recommender system really becomes useful. From all useful information it points out to the user which information other users find interesting and, at the same time, allows us to conclude that the recommender triggered the action.

It is easy to come up with several other scenarios. The reason for doing the exercise is not to list all possible scenarios but rather to show that users' subjective view of the social filtering functionality can be positive or negative for several reasons. It is only the third scenario above that can be caught by asking users. That is, users did not like what was recommended. On the other hand, it is possible that users do not appreciate (or use) the social filter even though it performs well in terms of modeling users (scenario 1, 4 and 5). Thus, how to evaluate filtering depends on what the evaluation sets out to measure. Subjective views are not enough to get at the filtering process, and likewise, pure functional analysis of the performance does not give the complete picture. If the filtering system works as intended but no one uses it, it could, for example, be concluded that the domain (or its users) is not suited for that type of filtering. One could argue that the recommender should only be used under the conditions of the last scenario and only filters information that is already deemed useful. It would act as a social filter on top of some content-based filtering processes.

In the recipe domain, there are a number of things to look for in order to determine whether or not the filtering process is happening. The obvious is to check if users choose recommended recipes or recipes that others have downloaded before them (i.e. that they at all act on recommendations). Furthermore, Kalas supports a number of ways to filter information based on content and it is possible to compare the number of recipes chosen from recommendations and recipes chosen from other ways of filtering information.

Pushing information to a user can have a very strong influence on people, and in a way, that is the purpose. Furthermore, users are not prone to scrolling, as it requires time and energy. For a user, it is convenient to look at the first 10 or so choices and choose among those. To evaluate the "push" effect of a recommender some other way of pushing information could be used, for example, randomly recommending recipes.

Subjective views and actual user behavior have to be crosschecked. We may find that users like the features but do not act upon them, which could indicate that social filtering serves more as a tool for inspiration and perhaps makes users feel less lonely in the space. In terms of the recipe domain, it is not unlikely that users would be interested in what recipes others cook from just for the fun of it. To summarize, key questions discussed in this section include:

- Do users see the social trails that other have left behind?
- Do the social trials help users filter out information?
- What are users reasons for acting on recommendations?
- In what ways do users filter out information?

5.3 SOCIAL TEXTURE AND QUALITY

Sometimes it is not enough that the information obtained is relevant, it must also possess qualities that can only be determined from how other users react to it, it needs a social texture. To follow up on the previous discussion, the social texture is what partly determines the usefulness of the information. To give an example, it is only when an expert verifies that a piece of information is valid, or when a piece of art is often referred to in the literature, will it be of high quality in the eyes of the information navigator.

It is important to understand the distinction between social navigation for filtering and social navigation as a tool for adding a social layer or quality to information. In the former case social information is used to (automatically) filter out relevant information, but the social information itself does not need to be made visible in all its details to users. The system could use the social information as a basis for recommendations without making users aware of it. In the latter case it is the overlaid social texture that makes the information interesting or useful. In terms of recipes, this could mean that both ingredients and recipe author serve to judge the usefulness of a recipe as a whole. Harper (2002) shows that this even applies to what might be considered a prototypical information retrieval scenario. He studied the work done by desk officers at the International Monetary Fund (IMF) who retrieve information concerning different countries and their economies. He found that a piece of information might very well be valid and important, and still completely uninteresting since the people with power in the country are not reading and acting upon it. In other words, he showed that the social texture could in many cases be the quality one is looking for when determining the value of some piece of information. In Kalas, this would translate into looking at who authored a recipe, where a recipe was first published, how many times it has been downloaded, who have commented it and so on.

With the same argument as for filtering this evaluation has to be carried out with care. The easiest way is to ask users if social texture is important when judging the quality of information. Though useful, the answers should also be verified against real user behavior. We might check for patterns in user behavior such as repeatedly choosing popular recipes or recipes from the same author. Generally speaking, is it possible to detect any patterns in the social information that surrounds the chosen recipes?

Social texture is not only a tool for making quality assessments about information. In Chapter 2, it was argued that there is more to navigation than moving from one place to another. We argued that navigation could be an exploratory activity. Navigation could support learning. In other situations navigation should be pleasurable and fun. Social navigation fits perfect with this aspect of navigation since it allows for social interaction to take place. In some circumstances these aspects of navigation can be the main driving force. The social activity becomes more important than the navigational activity. What once started out as a system for information navigation could transform into a system for both seeking information and other social trails users have left in space. This view of social navigation makes the social texture important in its own right. The social features thus improve the overall experience of the system. Chatting and other forms of communication between users are expected to be diverse and not only related to navigational problems. Thus, the social features do not necessarily help users find the shortest path in space, but instead helps them explore the space they are in.

These aspects of the social features can be evaluated by asking users if they appreciate them. An evaluation of the communication patterns in the systems will require a manual investigation of what is said in chat sessions. One such analysis was conducted in (Persson and Fagerberg, 2002) where postings were divided into several groups (test notes, meta-notes, expressiveness, information, comments, drama, and chat) and then the number of posting in each group counted to get a picture of what type of communication was present in the system. In systems where chatting is used extensively such manual analysis may not be feasible. In such cases it could be possible to conduct a simpler analysis by counting discriminating words.

If users stay longer because of the social features it could be argued that this shows how the system is used. It makes it possible to claim that the social feature is not just used to make the navigational process more effective, but also supports users with other activities such as exploration of the space. To evaluate how long users stay because of the social features would require two systems: one with and one without them. By comparing how long users stay in both systems the effects of the social features can be measured. It is also interesting to accompany such evaluation with users' subjective views on how long they spent in the system. As was shown in the Agenta & Frida evaluation (Höök et al., 2000) users often overestimate the time spent in a boring system and underestimated the time spent in a fun system (compare this to watching a good movie and a bad movie). This is an example where we want different results from the objective and subjective analysis. The discussion on social texture can be summarized in the following questions:

- Are users positive towards the social texture?
- Do users judge the quality of information based on social texture?
- Will the social features improve the overall experience with the system?
- What do the communication patterns look like?
- How long do user stay in the system?
- Do the social feature help users explore the space?

5.4 SOCIAL AFFORDANCE

To afford something means to "*to make available, give forth, or provide naturally or inevitably, the sun affords warmth to the earth*[15]". There are many objects in the physical world that afford manipulation in a certain way. By looking at a doorknob one instinctively knows how it should be used. A big green button on a Xerox machine affords to be pushed. This is true for the western world where the color green signals "go ahead" and red signals stop.

Similarly it is possible to imagine that a social situation or social information affords certain behavior. The visible actions of users can inform other users of the appropriate behavior, what can and cannot be done. Not only does social affordance induce certain behavior, it can also make people feel relaxed and comfortable in the space they are in. Social affordance makes a space feel more inviting. Consequently, social affordance does not necessarily help users to navigate more efficiently or help them to find exactly what they are looking for, instead it makes users stay longer in the space, it inspires users to try new functionality, or helps users understand the space they are in. Allowing users to see each other and their actions helps them to pick up on the 'norms' for how to behave in space.

To some extent social affordance overlaps with both social filtering and social quality and can be evaluated in much the same fashion. What is noted though is that social affordance is very difficult to observe without the use of some post-questionnaire analysis of the system under evaluation. To feel inspired or picking up norms is not something that is easy to observe from usage data.

There are, however, things that can be measured. In Kalas, we will not only look for the reasons for choosing recipes (e.g. if users choose a recipe based on its comments), but also ask users explicitly if they felt that they learned anything from the other users. From usage data it should be possible to make movement patterns visible, given that there are any (remember that such analysis is difficult and error prone). Are users inclined to go to recipe collections with many users in them? If any such patterns can be seen they would be indications of social affordance happening. If a user moves to a collection because there are many users there, the social situation is said to afford going to that collection to find good recipes. A plausible way of analyzing movement patterns is to set up a system such that users always start from a predefined place (in Kalas the "My Box" recipe collections) and analyze where they go from there. In many ways analyses of movement patterns

[15] Explanation from Mirriam-Webster (http://www.m-w.com/home.htm

touches on social filtering. If a user bases her information choices on where others are – she selectively chooses recipes from popular collections - the observable actions of others turn into a social filter.

In a more subtle way we expect the social situation to affect people. Can places in space become interesting because of the number of people visiting them? To speak in terms of Kalas: will the social situation affect users' sessions? Over time it should be possible to notice fluctuations in what collections users visit. By the same token as people tend to go to popular restaurants, we expect the same type of behavior in a system built up from recipe collections. Note that the collection or a restaurant does not necessarily have to change its contents to make this happen, it is the number of people in them that changes. In summary, social affordance stresses the following points:

- People attract people.
- Users learn from others in the system.
- Users are inspired to act in certain ways by watching others.

5.5 Usage reshapes functionality and structure

Social navigation design may alter the organization of a space. In amazon.com, the structure of the space experienced by visitors is changed: one can follow the recommendations instead of navigating by the search-for-terms structure. The very foundation and purpose of systems such as the CoWeb (Dieberger, 1999) is the ability to actively modify the space one is in. Instead of letting designers modify a system to fit the users' need, why not allow the users to modify the system? For many situations the one-size-fits-all metaphor is not adequate and who is better to judge what a user needs than the user herself?

Social navigation could be a first step towards empowering users to just this. In a natural, subtle way, it allows the functionality and structure to 'drift' thus making information spaces more dynamic. It is natural to ask if users see social navigation as a tool for changing the space and deliberately use it for that purpose. Arguably, this would work in the other direction as well; some users will not perform certain actions to make the system as stable as possible, for instance, to not influence a recommender system. This reasoning transfers into the following claim: some users deliberately act in a certain way to influence or not influence a system.

We want to stress the deliberate here. In social navigation users always modify the system. However, in most cases users are not aware of it, or at least, do not modify the system for some specific purpose. Take a history enrich environment for example. Whenever a user walks a certain path she will modify the system to make that path just a little bit more visible to other subsequent visitors. The users intent is not to make a deliberate *trail* in the system. On the other hand, we could imagine a scenario where another user walks the given path over and over because she deliberately wants other users to see and walk the same path as she. The phenomenon opens up interesting ways of using social navigation as a tool to enhance certain aspects of a system. It is not unlikely that some users try to push information to others or even try to make themselves more visible. In the recipe domain such behavior could manifest itself by users voting on the recipes they have added to the system over and over, to make others see them.

The issue of preventing the system from changing is also interesting. If a user is happy with her configuration and, for example, is satisfied with the way a recommender system models her, how can she keep the system in that state? One possibility is to stop voting, but if the system is set up in such a way that all actions affect it there is not much she can do to prevent the change from happening. As an example, printing recipes in Kalas renders a positive vote on that recipe, and consequently, updates the recommender system. It becomes even more interesting when considering actions that users want to reverse. Should users be allowed to erase their traces from a history-enriched environment? Allowing users to actively modify their traces potentially undermine the whole idea of social navigation. Remember that it is not only one user's view of the system that is changed, but also every other user's view. Still, it is tempting to give users more control to, for instance, handle snowball effects. We offer no answer to this dilemma. However, if an evaluation reveals that many users often try to influence a system, designers should consider adding more user control over how social information is used and manipulated.

How can deliberate reshaping of space be evaluated? By now it is obvious that the easiest way to evaluate social navigation is to ask users for their opinions about the system under evaluation and to evaluate how *usage reshapes functionality* is no exception. In fact, we argue that this is one of the questions most suitable for a post evaluation. The reason for this is that there is no apparent right or wrong answer. Compare the question "Did you try to influence the system?" to "Did you find the overview map useful?". In the latter case it is more obvious what the desired answer would be.

It should be possible to look for anomalies in the way users interact with information. To actively influence or push something often induces repetitive behavior. For example, commenting information over and over, being extremely active in chats, or downloading the same information repeatedly. This type of behavior has to be verified against users' subjective statements on why they acted a certain way. In Kalas there are numerous explanation to why users vote more than once on a particular recipe, where boosting it is one explanation. Simply deciding from usage logs that users exert certain patterns of behavior will only give a weak trend of what is happening.

If a designer wants to evaluate certain aspects of social navigation she may need to design the system in a certain way to make them easier to catch. To give an example, in Kalas we could have added new recipe collections during the course of the study, both collections that were created by users and collections that were derived from actual usage data (c.f. the dynamic recipe collections discussed in Section 4.2). Doing so would have allowed us to compare changes in usage. Then, if collections created from usage data were to be frequently visited, it shows that usage patterns can indeed be used to restructure a space. In summary, the key questions discussed in this section are:

- Will users deliberately try to modify the system?

- Will users deliberately avoid certain actions to keep the system in a stable state?
- What are the reasons for repeating behaviors?

5.6 Privacy

Social navigation relies on the visibility of people and their actions. Thus, people have to give up some of their privacy to make social navigation work. If we know in what circumstances and to what extent users are willing to do this, we can design better systems. In the EFOL study it was observed that most users were willing to share almost any type of information while others did not even want to be visible in a system. Thus, we want to explore one aspect of privacy that affects social navigation, namely invisibility. That is, what type of people use invisibility and how do such people behave in a social navigation system.

Intuitively, the reason for being invisible is because users do not want to share information. If that is the case privacy aware users will only share a minimal amount of information about themselves. They will not give out information such as age, their full name, gender, or email.

We argue that users who are willing to share information are socially active and that the socially active users will have fewer issues about the social features in a system. One possibility to find the socially active is to observe people act in the real world and transfer that knowledge to the virtual world (Munro, 1999; Harper, 2002).

Since the social navigation metaphor is grounded in how people behave in the real world, that behavior can be used to group people. With this reasoning we would first have to collect information on how users behave in general. A pre-questionnaire could contain questions such as *"Do you make contact with people you do not know in the grocery store?"*. However, there are differences in the way we behave in computerized environment versus physical environments, which could entail slightly different privacy concerns (Palan and Dourish, 2003).

In the Kalas evaluation light weight questions will be compared against the amount of voluntary personal information users give out, whether they are positive towards the idea of sharing recipes, if they are positive towards showing their actions to others, if they actively comment recipes, if they vote on recipes, and finally, if they chat. There are a number of ways to measure the degrees of privacy intrusion or reasons why users would want to be left alone. To some extent it is possible to verify this by analyzing logs (e.g. are users in general more invisible to start with?). However, those findings need to be backed up by post-questionnaire information. For example, a user could forget to turn the visibility flag back on after turning invisible. In Kalas, we try to minimize this particular risk by making it very clear in the interface that a user is invisible (she is not seen in the overview map and she is not able to read or write messages in the chat box). Claims that can be made about socially active people:

- People who choose to be invisible are less keen on sharing information than others and more negative towards looking at what others can do.
- Socially active people will share more information than the average person.

5.7 CONCEPT DRIFT AND SNOWBALL EFFECTS

Over time people and information change. What was interesting to a user today may be totally irrelevant tomorrow. A common example of this rapid concept drift would be news, which quickly gets outdated. Typically recommender systems and history-enriched environments do not cope very well with this shift in interest. They tend to get "conservative", meaning that once a user profile – or path - has been established it is difficult to change it. In order for social navigation to be truly successful it has to take into consideration that people's interests change and that different types of information have different expiration dates.

Social navigation is highly dependent upon domain and what works in one domain can be a failure in another. The very nature of concept drift (i.e. changes in preferences) makes it a problem that is largely dependent upon domain. It is difficult to give any practical advice on how to evaluate it in general terms. To illustrate this, the following paragraphs show how concept drift can manifest itself as a problem in the recipe domain.

On the surface, concept drift appears to be a minor issue in terms of food. A person's preference for food is stable and she is not likely to change her personal taste from one day to another. However, if we look at our preferences for different kinds of food over the year, the picture gets more complicated. During Thanksgiving we want some special food, during Christmas something else. What we like to eat during summer time is different from what we eat during winter. Thus, even though the individual preference for a user stays the same, what she wants to cook varies over the course of a year.

To show how this can become a problem, consider a recommender system - or any other system that models historic information. Around Christmas there are some recipes that should be recommended but only for a very short time. If many users use a recipe system for this purpose alone (i.e. to find recipes one usually does not cook) the recommender system's model of users can become very skewed. It will reflect a very specific dimension of a user's preference for food, namely her – more or less – predefined preferences for what to eat around Christmas. What could happen is, of course, that new users enter the system after such peak and are recommended cooked turkey. Thus, in the recipe domain we would have to evaluate the recommender system over a period of time and see how it handles these periods of very odd user behavior. We might derive a conclusion that forces us to build in some heuristics in the recommender to handle such peaks. Of course, if people only use the system for such special occasions then it is not a problem.

We began by saying that our preferences for food are stable and not likely to change. However, we argue that this is not the case. On the contrary, they can change in an instance. One extremely popular cooking show on Swedish TV[16] demonstrates this is not necessarily so. The chef, who is extremely fond of ginger and lime, is likely to have had a major effect on what type of ingredients people like. In the same way as in music and fashion there are trends in what type of food

[16] The Naked Chef, http://www.bbc.co.uk/food/nakedchef

people prefer. The point is that our preferences towards things are constantly changing and influenced in ways that is not always apparent and foreseen.

Does this mean that recommender systems do not work and that “most popular” indicators are useless in the recipe domain? First of all, it is our strong belief that it works in the domain; otherwise we would not have investigated it. To some extent a recommender should be able to pick up on fluctuations in user behavior and given enough time the history information will reflect what is actually popular. It is, however, probably the case that some users will be disappointed: users who are very sensitive towards trends and constantly change preferences. But, then again, some users could find the social information useful just to find out what “the masses” like and use the recommender as a tool to find the recipes not to choose from. To evaluate concept drift the following issues needs to be considered:

- Users’ preferences change over time. The degree of change varies between users.
- Concept drift is domain dependent
- Concept drift needs longitudinal evaluation to be seen

5.8 BOOTSTRAPPING

Throughout this thesis we have argued that social navigation works best under conditions where there are many available users. The reason for this argument is the *bootstrapping* problem. Namely, when there is no information to model a history-enriched environment, for example, works poorly, creating paths that are random and not a reflection of user behavior.

One solution to this problem is to combine collaborative filtering systems with content-based filtering techniques. However, there is more to this problem than to bootstrap recommender systems. There are many situations in which bootstrapping a social navigation system becomes an issue:

- In a system that relies on online awareness. How will the system handle the situation when there is very few people in it, that is, how does the system make the first few users remain in the system?
- When a new user arrives to an already populated system and the system does not know anything about that user, how can it match her against other users?
- In a system that relies on users actively leaving their trails (i.e. commenting information) how can this behavior be encouraged?
- When new information arrives in the system how is that information made visible when there is no social information surrounding it?

These are some examples where we face a potential bootstrapping problem. In reality, more or less every type of social navigation faces this problem. Of course, some situations are more sensitive to bootstrapping than others (e.g. pure collaborative filtering). Thus, one possible reason for why social navigation might fail can always be traced back to the bootstrapping problem. It is also a recurring problem. Old traces of users can be outdated, rendering them useless. In many systems, the social trails have to be constantly updated in order to avoid the problem. So, how is this issue detected and evaluated? We see two plausible avenues to examine. The first one is relatively straightforward and can be formulated in the following claim: "*The longer the system is used, the more the social features will be used*".

Thus, we implicitly state that bootstrapping is a problem and it will be less so when more usage patterns are added to the system. Keep in mind that for any system it is likely that its users needs some time to get used to the system before they start using the more elaborate features. By this we mean that there is a natural learning curve in any system. The natural way to verify the above claim is to do a longitudinal evaluation of a system to see if any changes in the way people use it when it gets "socially richer" can be detected. To give an example, we could analyze if users act more on recommendations as time passes.

In the other avenue we simply examine the reasons for why some of the potential benefits did not happen. To be precise, if a designer does not see the effects of filtering, the social texture and so on, she should look at bootstrapping as one answer to why she did not see the effect she had hoped for.

It is tempting to conclude that a particular feature did not work because there were not enough users in the system. One could always claim that: "*if we only had more users in the system the effect would have been seen*". This argument is dangerous because almost any negative evaluation can be explained by it. This is one important reason as to why we argue that large and longitudinal studies are called for: they rule out bootstrapping as the reason for failure. If a system is poorly designed, the social functionality will not be used (independent of the number of users), the longitudinal evaluation will show this, in effect, making us look for additional causes. However, in many cases social navigation can be evaluated in small-scale evaluations (see Section 4.4). Lastly, a designer should have some reasonable idea of the critical number of users that is needed to make social navigation work. It helps her to rule out bootstrapping, and to conclude that in a specific domain and situation social navigation did not work. Key issues and questions discussed in the section are:

- If an effect of social navigation cannot be seen the bootstrapping problem should be investigated
- The use of social features should be evaluated over time.
- What is the threshold of users that has to be present before effects of social navigation can be measured?

5.9 Design

Until now we have focused on the benefits and problems that can arise when using social navigation. Those effects can only be observed and made to happen if the system mediates its functionality in the proper way. In evaluating social navigation it is natural to ask the question, what is the best way to design for it? This entails deciding on which of numerous ways social texture should be communicated, such as: how to mediate the presence of other users, or finding the implicit or explicit actions that can be naturally included in the system dialogue and still be useful to infer users' preferences from. The way to evaluate the design will be specific for each system and context, thus it is difficult to discuss it in general terms. For example, the Babble system (Erickson et al., 1999) mediates user presence through its social proxy, as shown in Figure 5.1. The evaluation must be carried out on a case-by-case basis. With this said, there are design issues that are more or less the same regardless of the system under observation such as how to best visualize recommendations to users or how to mediate online presence.

It is preferable to evaluate a design in regard to the benefits and problems discussed above, ruling out design flaws as the reason for not observing the effects of the social features (compared to the bootstrapping problem just discussed). Let us just present three candidate questions to ask concerning the design of a social navigation system.

Question: "Does a system clearly communicate to its users that there is social information that can be used to judge the quality of the actual information?".

If it turns out that users do not use the social texture to judge the quality of information this question allows us to rule out possible design flaws in the interface.

Question: "Does the design allow users to naturally leave explicit or implicit traces of their choices behind that can be subsequently accumulated and presented in suitable ways as social texture on top of information?"

In a system with a focus on social quality aspects (i.e. enabling users to leave and make use of the social trails to make judgments about information), usage trails must be visible and easy to interpret. In Kalas, for example, this question focuses on how and if users leave comments, and if the system was designed to make that happen in a natural, fluent way.

Question: "Are users helped or disturbed by constantly seeing the effects of changing the system?"

In Kalas, users constantly change the configuration of the system by turning invisible, voting on recipes, and so on. This can put an additional cognitive burden on users. Will other user movements in the overview map be disturbing to some? What are the effects of the list of recipes constantly changing order because of the recommender system? How do users react when other user switch from being visible to invisible and back again? In short, a system that supports social navigation runs the risk of being a very cluttered and "flickering" system. It runs the risk of being perceived as unstable and difficult to navigate.

There are many more questions that we could come up with and evaluate. What we first and foremost want to stress is the importance of making the social features reachable and understandable, if they are to be used.

Figure 5.1: Babble and an explained recommendation at Amazon.com.

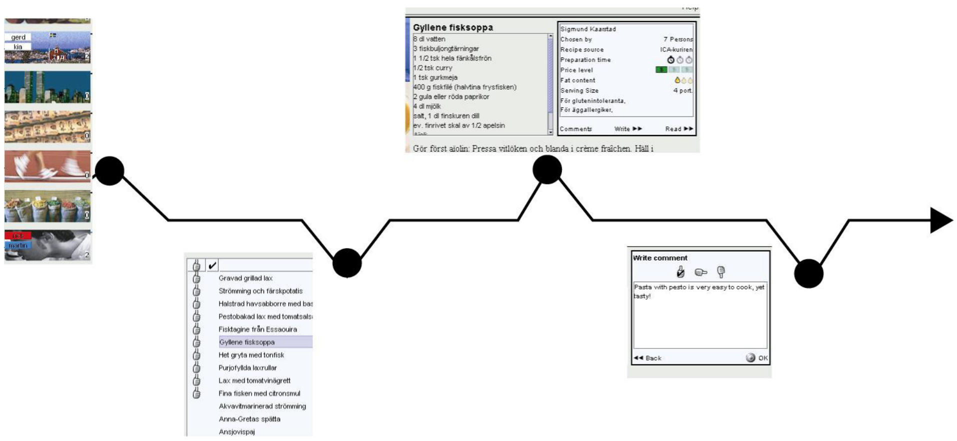

Figure 5.2: The navigational process in Kalas.

5.10 THE NAVIGATIONAL PROCESS

Each problem and benefit has been presented separately although it is apparent that they overlap. The benefits are not and should not be viewed as orthogonal axes in the "social navigation space". Many of the key issues can support the designer in evaluating several aspects, and the same goes for the problems (snowball effects and bootstrapping are two sides of the same coin).

Navigation is a process that involves several activities. Throughout the navigational process a user makes several decisions to arrive at her final destination. Social navigation is a tool that should support not one but several of those actives. Social Filtering is useful to help a user to formulate a destination or help her choosing a correct route. To speak in terms of navigation as defined in section 2.2, the social texture can help the user recognize that the destination has been reached. The social texture adds connotative information to a location. Figure 5.2 outlines the navigation process in Kalas. In the figure navigation is depicted as going from the left to the right. At several stages in the process social navigation will help the user in reaching her final destination. First by following the crowd the user finds an interesting recipe collection. With the help of the recommender system, the social texture, and finally, the comments she arrives at some recipe. It is a somewhat hypothetical situation and we expect other tools to help a user in the navigation process. It also gives the impression of navigation being very linear when it is in most circumstances highly iterative; a user is constantly reformulating her goal. Nevertheless, it demonstrates how the various aspects of social navigation come to play in various stages of the navigational process.

When evaluating social navigation the designer should look at the whole process rather than just focusing on one aspect of it. Is the navigational support well balanced and helpful to the users from formulating a goal to reaching the final destination?

As with the principles for design we do not want to argue that the questions and claims posed should be used exactly as is, in the same fashion as heuristic evaluation (Nielsen, 1990). In that sense it is not enough to get a group of social navigation experts to answer them. They are the articulations of how filtering, social quality, affordance, usage manifest themselves. If a system successfully adds a social layer it is expected that users will use that to judge the quality of information. To take another example; if social navigation works it can make users stay longer.

Social navigation is not easy to evaluate. The fact that it requires people to leave their traces is a major reason for that. Another is that it often requires many users to do so. However, it has already been shown that with small means it is possible to make useful and interesting observations, as was done in the study of EFOL. In some circumstances it is also possible to make do without any user group at all. To evaluate the performance of a collaborative filtering system a designer can often use existing data sets from earlier evaluations (see for example, the EachMovie[17]) with standardized metrics for prediction performance (Breese, et al. 1998).

---

[17] http://research.compaq.com/SRC/eachmovie/

6 EVALUATING KALAS

While the evaluation of EFOL served as a tool for us to redesign a system that uses social navigation, the Kalas evaluation will take the generalized ideas discussed in the previous chapter and apply them to a real world situation.

One objective is to evaluate social navigation in the food recipe domain. The EFOL study indicated that the domain, online food shopping, could be turned into a social place. Will this still be true when we move away from the controlled lab environment and scrutinize Kalas (and social navigation) under more realistic conditions? Also, were the issues raised in the EFOL evaluation (e.g. people's need to give more social input) dealt with properly?

Another objective is to describe, in detail, how such an evaluation can be carried out, what data sources are needed, what type of questionnaires can be used, and in what setting it should be carried out? What is the interplay between subjective views and actual user behavior and how do we handle conflicting results from various data sources? The intention is to present a case study for others to use as a guide for future evaluations of social navigation systems.

A third objective is to take a closer look at the difficulties in evaluating social navigation. We have said it many times: social navigation relies on people. What are the difficulties we face and what can be done to minimize the effects of them?

We do not expect to sort out all the issues in just one study, but many will be discussed and answered in part. The most important goal is to analyze the social navigation features in Kalas, and that is where the bulk of the chapter will be spent.

6.1 TASK AND PROCEDURE

The EFOL evaluation was a controlled lab study conducted on a small set of subjects (12 in total) who were computer science students. The pre-study was qualitative and mainly allowed us to improve the design of Kalas and identify certain issues that needed to be investigated further (see Section 4.5.1). In particular the pre-study indicated that:

- the recommender functionality was difficult to grasp,
- users wanted to give explicit feedback on recipes,
- users wanted to comment on recipes, and
- a minority of users was disturbed that other users could see them in the interface.

Our goal was to move away from the laboratory situation and get a large number of "real" users so that we could study Kalas under realistic conditions. Only then would it be possible to see the real difficulties in bootstrapping a social community around the system.

6.1.1 Recruitment of subjects

The Kalas study was conducted in close collaboration with the online cooking portal www.hemma.net[18]. They provided us with over 3000 recipes. hemma.net is a Swedish web site that has well over 10000 active users. The portal mainly consists of information related to the home (e.g. food recipes, refurbishment, and so on). It was therefore easiest for us to use hemma.net as a basis for recruiting subjects to the study. The recruitment was carried out in two different ways. Firstly, two emails were sent out to all subscribed users at hemma.net. Secondly, links were added to the portal and were regularly displayed as a news item for two months. Before entering the system users could read that the system allowed them to select recipes, save and print recipes and chat with other users.

6.1.2 Method

The evaluation took place over a period of six months, from June until November 2001 and it was open for anyone to participate. Subjects did not have any particular task to carry out and could use Kalas whenever they wanted, as much as they wanted. To use the system subjects first had to fill in a pre-questionnaire (where they also signed up for in-depth interviews). After five months, subjects were prompted to fill in the post-questionnaire. During the period of use all user actions were logged.

Four users were selected for in-depth interviews allowing for a qualitative evaluation of Kalas. The purpose of the interviews was to verify the findings in the post-questionnaire and conclusions drawn from log data. The in-depth interview allowed users to express their subjective feelings towards the system. The in-depth interviews were carried out over the telephone.

6.1.3 Data sources

The layout of each data source in the study (pre-questionnaire, post-questionnaire, in-depth interviews and logs) is gathered in appendix A. They were related to each other in the following manner. In the pre-questionnaire users filled in their email address and in the post-questionnaire they again filled in their email address and the login name they had used in Kalas. Consequently pre and post questionnaire could be matched using email address, and questionnaires and usage data by login names. This means that full correlation between data sources could only be achieved for those who filled in the last questionnaire.

Questionnaires and interviews were in Swedish and have been translated to English.

[18] The Portal is now closed in favor for http://www.icakuriren.se

The final point to make is the way the post-questionnaire was set up. The answers to the first four questions of the questionnaire determined the number of questions users had to answer. As an example, if a user did not recognize an online user she did not get any more questions relating to online users. The number of users who answered any given question will therefore vary.

6.1.4 SUBJECTS

In all we had 598 subjects completing the first questionnaire, 530 females and 68 males. Most subjects, 309, were between 30 and 50 years old. They were also frequent Internet users: 445 accessed it every day and another 129 two to four times a week. 225 subjects had some computer science related education and 89 of the subjects were skilled in cooking. Since subjects were solely recruited online we got people from all over Sweden, mostly women living in smaller cities. Of the 598 who answered the pre-questionnaire, 302 used Kalas (subsequently referred to as the *active* group). This involved downloading a stand-alone application written in Java, and installing it on your computer. Lastly, 73[19] subjects answered the post-study questionnaire (subsequently referred to as the *final* group). The groups are summarized in Table 6.1

This was an uncontrolled study, and as such, there was no way of forcing people to complete all steps in it. In the forthcoming analysis only data from the active and final user group will be used. Questionnaire answers (both for the pre and post questionnaire) will only be reported for those who completed the whole study, i.e. the final group.

For some users it was not possible to connect pre, post questionnaires and usage logs, and that is the reason why the number of pre-questionnaires (69) is slightly less than the 73 subjects who filled in the last questionnaire.

[19] 3 users who did not answer any of the questions in the post-questionnaire were removed.

User groups	Size
1. Subjects answering the pre-questionnaire	598
2. Subjects using Kalas	302
3. Subjects answering the last questionnaire	73

Table 6.1: Summary of subjects and user groups.

Pre-questionnaire question	M1	M2	P-value
1. I make contact with people I do not know, e.g. at the bus	3	4	.06
2. I ask someone for the way when I am lost	6	6	.52
3. I share my experiences with others	6	6	.51
4. When I am on holiday I make contact with people I do not know	5	5	.42
5. Use Internet[1]	11	11	**.04**
6. Use computers[1]	11	11	.08
7. Use chatting[1]	5	5	.49
8. How often do you use hemma.net?[1]	8	6	.07
9. How often do you cook[2]?	7	6	.21
10. What are your cooking skills?	5	5	.52

Table 6.2: Median scores for the final group (M1) and those who answered the pre-questionnaire (M2). Question 1 to4, 10 are on likert scale 1 (negative) to 7 (positive). The P-value measures any significant difference (less than .05) in distribution of scores between the groups. Significant differences are marked with bold face.

1. Answers were in text form and coded to numbers for the analysis (see Appendix A for coding procedure).

2. Answers ranged from 0 times a week to 8 or more times a week.

THE DROP OUT

296 out of 598 (close to 50%) only filled in the pre-questionnaire and did not move on to use the system. At first, this might seem like a very large number that poses a serious problem for the validity of the results. We argue that this is not the case and this mainly due to what it is that is actually under observation.

First, what are the reasons for the 296 not moving on to use the system? Mainly, it boils down to either of two cases either they (1) decided to not participate or (2) had problems installing and downloading the software. The first case need not to be explained any further, the study was set up in such way that users had the opportunity to drop out.

The second case appears to be a more serious problem. Again the nature of the study and the fact that Kalas is a research prototype inevitably leads to some users dropping out because they, for various reasons, could not start the system.

In a controlled laboratory study it is possible to have total control over who participates. In theory a controlled study should allow all types of people to participate. In practice, of course, this is difficult. For an open study this is certainly not the case. It can be expected that curious people who wanted to participate in a study signed up for the Kalas evaluation, independent of whether they moved on to use the system. This "self filtering" is likely to affect the result more than the drop out. We have already from the start a user group that is not reflected within the general population. The fact that subjects were recruited from one particular portal is also likely to affect the result more than the drop out. We have also seen that many more females than males completed the first questionnaire. On the other hand, we could claim that we are actually, by doing this, studying the very target group of future Kalas users.

To make sure that there were no apparent differences between the users who signed up for the study and those who completed the study, their answers to the pre-questionnaire were compared. In Table 6.2 a selection of the questions from the questionnaire has been evaluated with standard Mann-Whitney tests[20] (Siegel et al., 1988). First of all we note that there is only one significant difference, i.e. the computer experience, which would support the claim we made in our first argument. That is, Kalas may have been somewhat difficult to install and start, and the more experience with computers a person has, the more likely it is that she completes that step.

In conclusion, we are not saying that the drop out did not affect the result. A smaller number of users will possibly make it more difficult to see some of the effects of social navigation. Thus, the drop out should not contribute to any wrongfully drawn positive conclusions.

[20] The Mann-Whitney test is a non-parametric standard test for analyzing data on likert scales.

Pre-questionnaire question	Never	Less than once a month	1-2 month	Once a week	2-4 week	Every day
1. Use computers					5	64
2. Use Internet				1	9	59
3. Use chatting	34	20	3	5	4	3

Table 6.3: Pre-questionnaire answers on Internet experience.

Pre-questionnaire statement	1	2	3	4	5	6	7	Median (mean)
1 I Make contact with people* I do not know, e.g. on the bus	5	20	15	7	6	9	6	3 (3.59)
2 I ask someone for the way when I am lost		3	3	7	11	20	25	6 (5.7)
3 I share my experiences with others			2	5	16	27	19	6 (5.81)
4 When I am on holiday I make contact with people I do not know	1	6	11	12	11	15	13	6 (5.7)

Table 6.4: Personality types. 1 (never happens) to 7 (happens often). *One user did not answer the question.

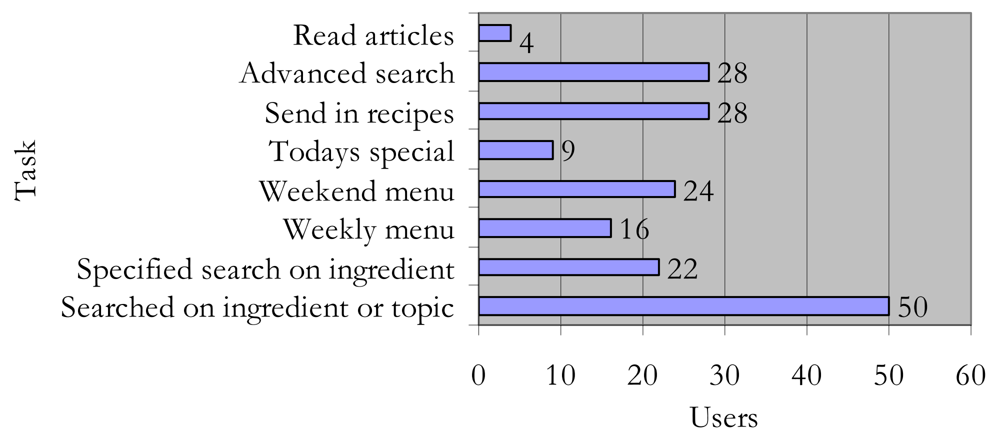

Figure 6.1: What functionality at hemma.net do you use? Multiple answers were possible.

6.2 Pre-questionnaire results

6.2.1 Personalities

Because of the nature of the evaluation there was no control over who signed up. Inevitably that could lead to a user group that did not reflect the general population. For that, some randomized procedure is called for. We expect to get a certain type of users in the study, those who are outgoing, curious, and, in general, positive towards participating in studies. The demographic and sociographic data, henceforth, can give a hint of what type of users we are dealing with.

We have speculated that social navigation will be most accepted by users that are outgoing and socially active. They are the ones who will actively contribute social information and have little or no problem being visible to others. The pre-questionnaire (see Table 6.4) contained four questions that related to how users acted in certain social situations. The group was open to the type of contact making we expect them to perform in Kalas. They are the type of users who make use of social navigation features in the real world (e.g. most of them would ask someone for the way if the were lost). Users are least inclined to make contact when there is no apparent reason to do so (Table 6.4, question 1).

The users should not to be intimidated by the social features found in Kalas. They are not expected to be disturbed by others trying to chat with them about recipe related issues. Arguably, if someone often asks for the way when she is lost, the same person should not be disturbed if others do the same.

6.2.2 Internet and computer experience

For obvious reasons the user group should be experienced Internet users and computer users; they signed up for the study online and installed a piece of software. In Table 6.3 this is confirmed. However, there is one thing that stands out. The users very seldom chat online, if ever. This is interesting and to some extent contradicts the type of users they are, outgoing and keen to make contact with others. On the other hand, the result does not say anything about users' attitude towards chatting. It could be the case that most users have not come across chatting or had any reason to chat online. Another possibility is that chatting - in the culture one is in - could have a negative connotation.

6.2.3 Use of online services for food searching

We have argued that it is not necessarily the case that users have a special goal in mind; they can explore a space or meet people. Hence, we wanted to establish how users searched for food online before they started to use Kalas. Did the typical user often use hemma.net (the site from where they were recruited) and what was her preferred way of finding interesting recipes? Users who choose the daily special (which essentially is a recommendation) ought to be positive to the recommender functionality in Kalas. Conversely, if users typically are advanced searchers who worked with food, it could be expected that the content-based search functionality would be used more often.

Figure 6.1 shows eight ways of using hemma.net. Users' preferred way of finding recipes is to use simple search queries. A substantial number of the users send in their own recipes (28 out of 69). Again, this is an indication of a socially active group of users that wants to reshape the space they are in. Users are open to choosing recipes based on some pre-filter (today's special, weekend menu, and weekly menu), a result suggesting that collaborative filtering is a useful tool in the recipe domain. It is clear that users prime reason for visiting hemma.net is to find recipes; only 4 users claimed that they read articles (i.e. not recipes) while visiting the site.

6.3 The active and final group

It could be claimed that only usage data from the final group should be used in the evaluation, but there are several interesting lessons to be learnt from those users who did not follow through the entire test period. There are reasons to look for differences between the active group and final group. If differences in usage can be observed it has to be taken into account when cross-validating usage logs with questionnaire answers. Furthermore, it tells us something about what type of user it is that completes this type of uncontrolled study.

Through the entire analysis, log data is used to cross-validate questionnaire answers. Logs are used to find trends in how users act on recommendations, when they turn invisible, if they make comments and chat, to name a few examples. It is reasonable to ask if there are any significant differences between the *active* group and those who completed the whole study. To rephrase, is it possible to use all log data in the analysis or can only logs from the *final* group be used? Several tests were carried out on user behavior that was deemed crucial in the forthcoming analysis. One-tailed Mann-Whitney tests were conducted, in which it was hypothesized that the final questionnaire group was more active than the rest of the users. Specifically, the following questions were asked (see Table 6.5):

- Are there any differences in the number of days users were logged in? The test indicated that users who answered the last questionnaire did spend more time in the system ($p < .01$).
- Are there any differences in the number of times users were logged in? The test indicated that users who answered the last questionnaire did visit the system more frequently than the other users ($p < .01$).
- Are there any differences in the number of times users became invisible? The analyses showed no difference between the groups (however only 33 users in total turned invisible). It is worth noting that the active user group had been invisible slightly more often. For reasons that will be explained later this behavior does not contradict what is expected.
- From the previous point it could be hypothesized that at least some of the people in the final group are more privacy-aware than the rest of the users. Is this consistent in the amount of information they share with others? When looking at how many times each user edited their "public profile" (i.e. name, email, description) there is a strong indication ($p < .01$) that users in the final group are more willing to share personal information. From this initial analysis the number of

times a user is invisible is not an indicator of how much information they share with others.

- The final group users were more active ($p < .01$) posters in the public chat space. For the same reason as with invisibility this test really only shows a trend, the number of posts being so small that it is difficult to draw any conclusions from them.

TYPICAL AND EXTREME USERS

Another way of looking at various user groups is to present users from each group. To get a feel for how active users were, how many times they logged on, and so on, a short summary of three types of Kalas users is presented below:

The average active group user. The average user is calculated as the mean of all actions from the active group. The average user used the system on 2.9 different days, she printed 1 recipe, saved 2 recipes, added 0.8 recipes to the shopping list, and gave explicit positive feedback on 0.97 recipes. Thus, in total she voted (implicitly and explicitly) on 4.77 recipes. Furthermore, she turned invisible 0.17 times and edited her profile about 0.6 times (setting age, email, name, and description), and lastly, she visited around 15 recipe collections.

The average final group user. For the final group the average user looks slightly different. This user used the system for 5.8 days, visited 29 collections and voted on a little over 13 recipes (printed 2.8, gave explicit positive feedback on 3.15, saved 4.5, and added 2.2 to her shopping list). She edited her profile 1 time and was invisible 0.2 times. The final group user is more active on all accounts compared to the active group user.

The extreme user. The most active user is female; she used the system on 31 different days and logged in for 33 sessions in total. She marked 17 other users as her friends and posted a total of 31 messages. This extreme user edited her profile over 60 times and changed collection 149 times during the period of use. The user voted on 20 recipes out of which 3 were explicit positive feedback.

In summary, users who completed the study were more active in the system. In the following discussion it will be clear when data from the active group is analyzed and when data from the final group is analyzed.

Usage data	Median (mean) final group	Median (mean) active group	P-value
1. Number of days logged in	4 (5.79)	1 (2.22)	**< .01**
2. Number of edit of user profile	1 (1.03)	0 (0.5)	**< .01**
3. Number of messages posted	0 (0.77)	0 (0.05)	**< .01**
4. Number of session with the system	4.5 (6.45)	2 (2.62)	**< .01**
5. Number of times invisible	0 (0.21)	0 (0.17)	.70

Table 6.5: Log comparison between the final group and active user group. Mann-Whitney test was used to measure any difference between the groups. Significant differences ($P < .05$) are marked with bold face.

Usage data	Count	Percentage of total
1. Number of recipes that got feedback (explicit and implicit)	1129	100%
2. Number of recipes recommended recipes that were chosen	199	18%
3. Number of recipes search for recipes that were chosen	305	27%
4. Number of recipes visible but not recommended that were chosen	90	8%
5. Number of recipes found by other means that were chosen	535	47%
6. Number of recipes that got explicit feedback	110	100%
7. Number of recipes recommended recipes that got explicit feedback	31	28%

Table 6.6: Summary of the recipes that got positive feedback in all collections but "My Box". Feedback is either implicit (print, save, and move to shopping list) or explicit ("thumbs up").

Usage data	Users	Recipes
1. Implicit feedback	157	409
2. Explicit positive feedback	56	171
3. Explicit negative feedback	7	11

Table 6.7: Number of users that gave explicit and implicit feedback. For each user the same recipe will only be counted once for each type of feedback.

6.4 Results

In the remaining sections of this chapter we look at Kalas in terms of filtering, social quality, social affordance, how usage reshapes functionality, privacy issues, and design, as outlined in Chapter 5.

6.4.1 Filtering

Users voted (i.e. gave feedback) on 1129 recipes during the period of use. Out of those, users voted on 199 (18%) recommended recipes (see Table 6.6 for a summary). Let us provide some context to help understand whether 18% chosen recommended recipes means that this social trail has an impact on the filtering process.

Predictive accuracy metrics measure how well the recommender algorithm can predict, from a sample set of votes, the future votes of a user. At a first sight it would seem that such an evaluation would be appropriate to evaluate the recommender functionality in Kalas. However, the predictive accuracy metrics tell us more about the underlying collaborative filtering algorithm than some qualitative aspects of the whole system. The actual algorithm that we used has undergone empirical tests by others e g (Breese et al., 1996), and yet another evaluation of the actual algorithm is not of interest here.

In addition, testing the algorithms predictive power implicitly assumes that an objective value can be assigned to the items (be it recipes, books or movies) in the application: they are either right or wrong for users. Instead, it is sometimes the case that a lot of recipes fit more or less with the user's taste and any of them can be chosen. What is needed is a means for the user to convince herself that it is a good choice, in particular, in the view of others.

Out of the 1129 recipes that got feedback 305 (27%) were found through the search functionality, and 199 (18%) were found from recommendations. The remaining 535+90 (53%) recipes were found through scrolling the list beneath the recommended recipes (see Table 6.6). Thus, the recommender system performs well in comparison to explicit search, which is interesting given that recommendations are not explicitly sought. Removing the recommender system would mean that users find 18% fewer recipes, or have to find them by other means.

It is known that users are quite influenced by what they can see on the screen, and scrolling down to see more recipes requires more energy than just choosing among the ones visible in the window. We therefore checked how many of the 18 recipes shown in the visible part of the window were selected. It turned out that the last 8 visible recipes (the ones not recommended) accounted for 90 of 1129 selected recipes. Thus, proportionally more of the recommended recipes were chosen compared to the visible but not recommended recipes.

It could be argued that the implicit feedback (such as printing the recipe or saving it temporarily) accounts for a large number of the chosen recommended recipes and thus the 18% selected recipes may not accurately reflect users' views on the recommendations. If the same comparison is made with implicit votes removed, 31 out of 110 explicit votes were on recommended recipes. That is, approximately

28% of the explicitly valued good recipes were recommended, meaning that when only looking at explicit positive feedback the recommender performs even better.

It is important that users at all understood that some recipes were recommended to them, and whether they found those recipes to be "good". The users answered the question: "*How do you think Kalas chooses recipes for you?*". 34 users (out of 73) believed that other users' choices affected the recommendations. From Table 6.8 (question 10) it is clear that users liked the recommendations, 69 users felt that the recommendations were good or very good. As a comparison only 3 people stated that they did not like the recommendations.

A user can enter Kalas for a number of reasons and with a range of different tasks. Some might want to cook something with some special ingredient and will be less inclined to limit themselves to only the 10 recommended recipes. Others will be looking for a particular style of recipe, e.g. spicy. Some might be looking for a recipe that can be quickly cooked. Thus, in many cases, even if the users' general preferences are captured by the recommender system, the recipe that fits with their "need for the day" will probably not be found among the recommended recipes (c.f. concept drift in Section 5.7). We would therefore expect that only a small portion of the chosen recipes to come from the recommended recipes, since that basically match a situation where the user's task is to be inspired and find any recipe. In light of that, it is promising that 18% of the chosen recipes were recommended and proves the value of having a recommender system alongside the other means of finding recipes.

VOTING

Our recommender relies on users voting (implicit and explicit feedback) on recipes. It is therefore interesting to investigate the relation between implicit and explicit feedback (see Table 6.7). In total 157 (52%) of the users gave implicit feedback on 409 recipes. The most common implicit vote was to save a recipe (416), followed by printing recipes (245), while saving a recipe to a shopping list accounted for 158 of the implicit votes. 56 (18.5 %) users gave explicit positive feedback on 171 unique recipes. In total, almost 21% of our users gave explicit feedback, which shows that it is a valuable mechanism.

In the study, seven users gave in total 11 negative votes. This is a small number and one might be inclined to draw the conclusion that negative feedback is not needed. However, six of the negative votes were preceded by an implicit positive vote on the same recipe – a way of correcting the system's profile about users. Thus, we argue that negative votes are particularly important in systems that build user profiles from implicit votes. Allowing for negative votes also makes it possible for users to provide *delayed feedback*: once they have cooked the food, tasted it and found that they do not like it, they can come back to the system and make their opinion explicit[21].

[21] The design of Kalas was such that when users re-entered the system the first place they went would be a "recipe collection" consisting of all the recipes they had chosen last time. Delayed feedback should therefore be fairly easy to perform in Kalas.

BOOTSTRAPPING THE RECOMMENDER SYSTEM

A recommender system should improve over time; the more input people give, the better the system should perform. In the ideal case it could be expected that users would choose more and more recommended recipes as the recommendations improve. To evaluate this, two cases were chosen: users that used Kalas at least two times and users that used the system at least three times. Only users who chose recipes in each session were included. The test was run over the active group and the final group. No tests yielded any correlation between number of sessions and chosen recommended recipes. If any trend was observed, it was the opposite of what was expected: users seemed to choose fewer recipes based on recommendations as they became more experienced with the system. Is this to say that the recommender system did not work? A plausible explanation is that the recommender system is of more use to newcomers. The experienced users will still act on recommendations, but also use other means of finding the recipes they need.

6.4.2 SOCIAL TEXTURE AND QUALITY

The social texture on recipes consists of the recipe author, how many times it has been downloaded, the source of the recipe, and its comments. Table 6.8 (factor 1-5), implies that the social texture is not that important when choosing a recipe. The more explicit social texture - comments and recommendations in chat - looks to be more important than other social information attached to recipes. In fact, 20 users out of 67 rated comments as an important factor, while only seven users claimed the recipe author to be an important factor. Within the social information we find large differences in terms of value. Both comments and chatting are in a way personal and directed to others, which could account for the large difference. It is interesting to note how the *number of times downloaded* marker is made very visible on Internet systems such as download.com and tucows.com. This could be an example of how the important social texture varies from one domain to another.

A series of questions were asked (6-9 in Table 6.8) to determine users general attitude towards the chatting possibility, the ability to comment, the ability to vote on recipes, and the ability to see other users. Users' answers indicate that they are positive towards the social texture on all accounts. Once again comments and chatting get the highest rating. Both these questions are targeted directly towards recipes, which makes it easier for users to see how they can be of benefit. Users like the ability to be social but do not necessarily base their recipe choices on those features. We argue that these functions made Kalas more pleasurable to navigate. They enhanced the users' experience with the system, although they did not explicitly affect users' choice of recipes.

CHAT AND COMMENTS

As the questionnaire answers indicate, people are generally positive towards introducing social features in Kalas. Is this view reflected in the way users actually behave, do they chat and make comments? Despite having 302 active users, too few were logged on simultaneously to allow for any serious chatting to take place. In all, six users made in total 10 comments and 15 chatted. It indicates that users like social features (i.e. subjective views), but do not necessarily use them (i.e. user behavior). One reason for not using the functionality is the users themselves. Only a fraction of them claimed to chat on a regular basis (see Table 6.3). In other words, the user group as such was not likely to chat at any great length.

When comparing the socially active people to those who did not chat or comment an interesting pattern emerges. In Table 6.9 this pattern is laid out. Users who chatted were more negative towards it (3.43 mean), than those who did not chat (4.16). For the ability to comment the result is more in line with what we expect, those who made comments are at least as positive towards the feature as those who did not.

This somewhat contradictory result can be explained by the nature of communication. Chatting is synchronous and dependent on a substantial number of people to work, which Kalas did not have. Comments, in contrast, are an asynchronous form of communication that are not – on the surface – affected by the number of users contributing. The analysis of the chat messages supports this explanation, as messages where mostly about getting someone else to chat, e.g. "*garemo, anyone there*"; "*hello*"; "*I have been kind of lonely in kalas [...]*". On the other hand, all comments were about the recipes, how to improve them, what ingredients could be replaced, and so on. As an example, one user made the following comment: "*The plum pie was very good and easy to cook [...] This pie is probably useful for other fruits as well, like peaches*". Another user gave feedback on some problems with the system via the comments, she said: "*Not all text is shown about how to cook the recipe when printing the recipe*". This way the comments act as a tool for the designer to improve the system and as a tool for users to help each other.

If we introduce synchronous communication to a system we have to make sure that there is a sufficient amount of people available to participate. In a system like Kalas, in which the feature has a very prominent role in the interface (c.f. to comments which are hidden until a user actively chooses to view them), this is, of course, even more important.

Post-questionnaire question	1	2	3	4	5	6	7	Median (Mean)
1. How important is the recipe author in your choice of recipe?	43	7	7	9	3	3	1	1 (2.1)
2. How important is the recipe source in your choice of recipe?	35	14	5	12	6	1	0	2 (2.2)
3. How important is it that others have chosen the recipe in your choice of recipe?	35	13	6	13	4	2	0	2 (2.2)
4. How important are a recipe's comments in your choice of recipe?	11	12	12	12	12	8	0	3 (3.4)
5. How important is it that a recipe was recommended in the chat in choosing a recipe?	13	11	12	11	8	7	1	3 (3.2)
6. What do you think about the possibility to comment recipes?	2	2	2	21	12	16	12	5 (5.0)
7. What do you think about the possibility to chat?	2	5	4	36	10	3	3	4 (4.1)
8. What do you think about the possibility to see others?	1	1	2	28	8	4	2	4 (4.3)
9. What do you think about the possibility to vote on recipes?	2	1	1	23	14	15	5	5 (4.8)
10. What do you think about the recommended recipes?	0	0	3	29	20	18	2	5 (4.8)

Table 6.8: Social texture and quality. Scores from 1 (negative/unimportant) to 7 (positive/important).

Post questionnaire question	Posted	Did not post	P-value
1. What do you think about the possibility to comment recipes?	5.6	4.98	.21
2. What do you think about the possibility to chat?	3.43	4.16	.07

Table 6.9: Difference in answers for those who commented and chatted and those who did not. No significant differences ($P < .05$) were found using the Mann-Whitney test.

Post-questionnaire statement	1	2	3	4	5	6	7	Median (Mean)
1. I learnt something from the others that used Kalas	18	10	5	8	2	2	1	2 (2.48)
2. I would choose a collection where there are interesting users	17	7	4	12	6	1		2 (2.2)

Table 6.10: Social affordance. Scores ranging from 1 (strongly disagree) to 7 (strongly agree).

Post-questionnaire statement	1	2	3	4	5	6	7	Median (Mean)
1. I deliberately tried to influence the system, for example, by choosing the same recipe over and over again	47	12	6	6	1		1	1 (1.7)
2. I avoided doing some things, for instance, choose a recipe, since I did not want to influence the system	51	4	6	9	1	1	1	1 (1.8)

Table 6.11: Reshaping space. Scores from 1 (never happened) to 7 (happened often).

Post-questionnaire question	1	2	3	4	5	6	7	Median (Mean)
1. How do you feel about the fact that others can contact you in the chat?	0	4	4	14	8	7	25	6 (5.4)
2. How do you feel about the fact that others can see your movements?	2	3	7	19	7	10	23	5 (5.1)

Table 6.12: Visibility and chat. Scores from 1 (uncomfortable) to 7 (comfortable).

6.4.3 SOCIAL AFFORDANCE

Watching and studying other users to learn something requires that people meet, but the user group was too small for this to take place on a regular basis. Given this, social affordance was hard for users to see or feel happening. Table 6.10 confirms this belief; most users claimed not to have learned anything from the other users in the system. For the same reason users did not go to collections with "interesting" users in them: there were simply no interesting users around for most of the time.

There are no clear lines between social affordance and social quality (see Section 5.10). We could argue that more or less every factor in Table 6.10 measures social affordance; users act in a certain way because of someone's actions. However, we want to claim that social affordance is more than a social texture. It makes users understand and learn something about the system. The fact that users feel they have learnt something from other users is not the same as saying that others influenced them.

From one perspective social affordance did not happen in Kalas, users did not feel influenced or helped by other Kalas users. From another perspective we want to make the claim that the evaluation did not allow the study of social affordance. The small number of users did not create the type of community that is needed to see the type of effects we were looking for.

6.4.4 PEOPLE ATTRACT PEOPLE

Did people attract people? The question tries to pin down what influence users have on each other in a more implicit way. This dimension of social navigation can measure both social affordance and social filtering.

When a user starts up Kalas she enters the system in the "My Box" collection, regardless of what she did in the previous session. Are there any particular collections a user usually turns to when they move from "My Box", and in what circumstances is this behavior changed? It was argued in Chapter 5 that one way of observing social affordance and social filtering would be to look at this type of pattern. The procedure was simple: all users' first moves were filtered out and compared against the number of people visiting the target collection. If a pattern could be observed in which users typically move to collections with many users in them, it would be an indication of people attracting people.

Two cases were selected: sessions with at least five users logged in and sessions with at least 10 users logged in. Moves were grouped as follows:

- moves to the collection with most users in it,
- moves to the collection with at least one user in it, and
- moves to a collection with no users in it.

In the first case (at least five users) 27 moves went to empty collections, 61 to collections with at least one user, and 23 to the most populated collection. In the latter the result was divided in the following way: seven moved to empty collections, 36 to populated collections, and 12 to the most populated collection. The pattern is fairly stable in both conditions, with slightly more movements to the most populated collection than to an empty collection. Both cases indicate that users go to populated collations rather than empty ones. The analyses, however, does not say anything about the reason for doing so. Still the most common pattern is to move to a collection that is somewhere between the empty and most populated one. It is interesting to note that many users (41) agreed to the following statement *"I prefer collections with few people in them"*. Thus, this could be an example of the unconscious influence users have over each other.

To evaluate if users attracted each other there have to be a minimum number of users co-present in the system. In order to draw enough attention to any given collection because of the number of users in it, there has to be a substantial number of users online. Consider the following example. If a person has to choose between two restaurants - one with no customers and one with 10 customers - it does not really matter which to choose. If, on the other hand, a restaurant had 15 customers and the other 100 customers that would likely have a stronger influence on the choice of restaurant. This did not happen to the extent we had hoped but there is a positive trend.

6.4.5 USAGE RESHAPES FUNCTIONALITY AND STRUCTURE

No users claimed to deliberately try to influence Kalas or deliberately avoid certain actions (see the answers in Table 6.11). With an overwhelmingly strong result like this, we might wonder if those few users who actively tried to modify the system acted in a certain way. Is it possible to find any patterns in their user behavior from log data? Just looking at the logs indicates that those users have much the same usage patterns as the rest of the user group (they had the same pattern in voting, selecting recipes, and giving out personal information). The one thing that could be observed was that the two users who answered a 5 and 7 on the first question in Table 6.11, both gave explicit feedback.

In Kalas users do not deliberately try to influence the system, at least not in terms of boosting particular recipes. If we elaborate on this a little further we find several possible reasons for why this is the case. First, users do not see voting on recipes as deliberately trying to influence the system. Second, the question that was asked might have led users in the wrong direction with the explicit example given, i.e. "*…by choosing the same recipe over and over"*. Third, they do not see, for example, making comments as influencing the system. This point goes hand in hand with the first one. It boils down to what is meant by *influencing* the system, and in particular, how to articulate that in a question so that a user can interpret it in the proper way. Last, and most probable, there are not that many apparent ways of influencing the system. We know from the pre-questionnaire that users upload recipes to hemma.net (see Figure 6.1), an action that reshapes the system. When there are no obvious ways to influence or not to influence a system, users do not see themselves doing it. Moreover, some actions can spawn other actions. Allowing users to upload their own recipes would likely affect other ways of reshaping Kalas. In that scenario we would expect to see more users trying to "boost" recipes by voting on them repeatedly.

6.4.6 Privacy

Table 6.12 reflects users views towards privacy issues. For most users it is not a problem that others can see them or that they can be contacted in the chat. In the pre-study two out of 12 users felt uncomfortable about being seen. In the study, about 17% felt uncomfortable or very uncomfortable by being seen. 13% of the users were uncomfortable with being contacted in the chat. In the pre-study it was the other way around, people were more negative towards chatting. Why this is the case we can only speculate; it could be that the subjects in the pre-study knew each other or that chatting is such an expected feature in online system that most people have learnt to accept it. Still, there is a group of around 17% that would probably benefit from a different design of Kalas. These results were crosschecked against those users who were invisible without finding any correlation.

There was no correlation between sharing personal information entered into the profile and users' degree of invisibility. Socially active users were also more invisible than the less socially active users (see Table 6.13, row one and two).

In both the EFOL study and Kalas study we have seen various ways in which privacy matters. Some feel uncomfortable that others can contact them in a chat while others do not want to be visible in an overview map. So far we have not looked at what type of personal information users are willing to share. Users were more or less equally willing to give out their age (120), email (113) and name (116). Only 22 users filled in an additional description about themselves. (All figures based on the 302 active users). We believe that this large difference is due to the nature of the description; it is seen as "optional". We did not expect that age would be the one personal information that most users would be willing to share. One explanation for this is that age could serve as a reasonable good filter to find users that share common interests. Another would be that providing age is extremely simple and something that users often provide in other situations.

Privacy and personality

A reasonable hypothesis is that people who turn invisible are less disposed towards making contact with people they do not know. As can be seen in Table 6.13, row three and four, there is a trend, though no correlation, that users who turned invisible are less inclined to make contact with people they do now know and ask people for the way when they are lost.

Usage data and pre-questionnaire statement	Invisible	Visible	P-value
1. Number of items in the profile (final group)	2.56	1.79	.82
2. Number of items in the profile (active group)	1.88	1.11	**< .01**
3. "I make contact with people I do not know, e.g. on the bus"	2.56	3.53	.06
4. "I ask someone for the way when I am lost"	5.22	5.72	.07
5. "I share my experiences with others"	5.56	5.77	.23
6. "When I am on holiday I make contact with people I do not know"	4.56	4.66	.46

Table 6.13: Comparison between those who turned invisible and those who did not. A P-value less than .05 is a significant difference between the groups. Significance is marked in bold face.

Post-questionnaire question	1	2	3	4	5	6	7	Median (Mean)
1. How easy/difficult was Kalas to learn?	2	2	2	20	18	12	18	5 (5.1)
2. How easy/difficult was Kalas to use?	4	1	7	19	15	14	13	5 (4.8)
3. It flickers on the screen when other users change collection, how do you feel about that?	1	5	6	17	4	2	11	4 (4.5)
4. If you noticed that the order in recipe list changed from time to time, did it disturb you?	1	2	6	12	1	1	2	4 (3.8)

Table 6.14: Design. Scores from 1 (easy/not disturbing) to 7 (difficult/very disturbing).

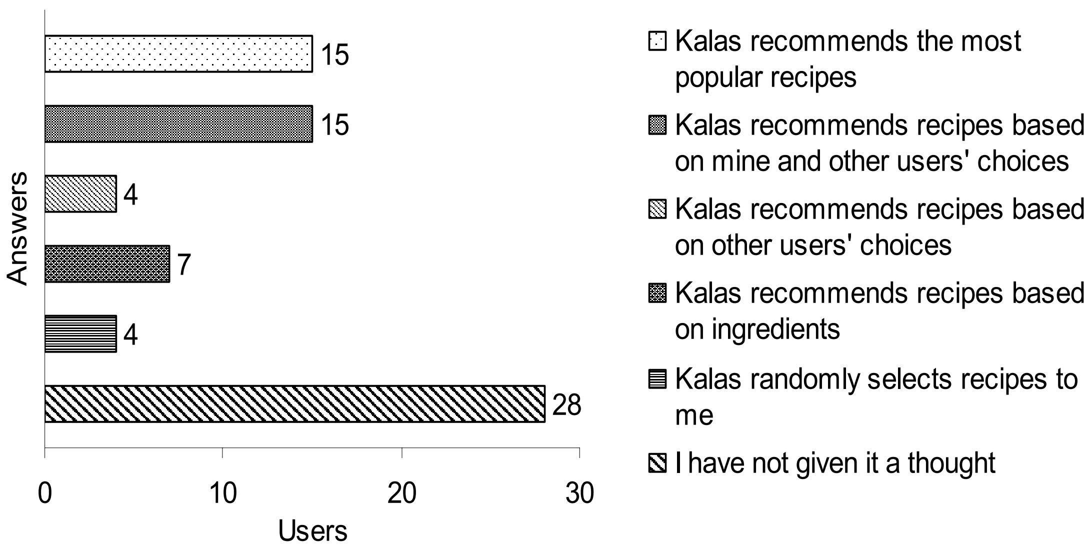

Figure 6.2: How does the recommender pick recipes?

6.4.7 DESIGN

Overall people found Kalas easy to learn (only six people found it more difficult than average). Similar results hold for using the system (Table 6.14, question one and two). Subjects understood the major social features in Kalas:

- 64% recognized the symbol for an online user,
- 88% were aware of the chatting facility,
- 92% knew that it was possible to comment on recipes, and
- 83% understood how to vote on recipes.

One figure needs further explanation. The lower recognition of online users (64%) is likely an effect of most users being alone in the system. The question users answered was exactly: "Is [picture] a symbol for another user?". While many users did not see any other user online, some of them did not make the connection that they themselves looked and appeared in the same manner as other users.

Real time awareness enables users to follow each other's movements in a system. A system that supports real time awareness will repeatedly be updated to show where users currently are, something that could potentially be disturbing. However when asked, our users responded that other users' movements did not bother them much: 12 users felt disturbed by it while 32 were not bothered (see question three in Table 6.14). As many as 27 users did not notice the feature at all, which is probably due to the fact they were alone in the system.

Concerning the recommendations we got a positive and quite surprising result. To the extent that users at all tried to understand where the recommendations came from, they understood that they stemmed from other users (see Figure 6.2). This result is completely different to what was indicated in the EFOL study where subjects had no clue of why they got certain recommendations. Our explanation of this result is the integrated use of the "thumbs up" symbol. The "thumbs up" symbol was added to the interface and placed next to each recommended recipe (see Figure 4.2). It was also used to allow users to vote for a recipe and thus placed next to the line "Good recipe!" (see Figure 4.3). We also used the symbol in the "my box" recipe collection that users first enter when they login to Kalas. In "my box" all the choices made at previous occasions are listed (see Figure 4.4) and the user can choose to change their opinions about some recipe, changing the thumbs up to a thumbs down symbol. This allows for easy delayed feedback.

A simple marker, like the thumbs-up symbol, make people aware of recommendations even if they do not necessarily explain the inner workings of the recommender, as was tried in (Herlocker et al., 2000). This relieves the demand to provide explanations of recommender systems – in our view, it is not necessary to provide the whole story of what is going on inside the recommender. It suffices to know that the recommendations are computed from the likes and dislikes of others.

Chatting and commenting have already been discussed from a social navigation perspective above. For chatting to work there has to be a critical number of users in the system. Although many users recognized both functionalities, we might ask if they could have been designed any differently to improve their functionality? The chat window was very visible in the interface and arguably easy to reach. In retrospect, we could have designed it slightly different and not restricted chatting to the recipe collection level, but introduced it as a community chat (c.f. EFOL). The chosen design required users to be in the same collection to be able to chat, which, for a small community, makes it harder to happen. The conclusion is that as the community gets smaller chatting should be allowed at a system level rather than at a "collection" level. Concerning comments they could have been more visible in the interface, by not obliging users to click a button in the recipe to view them.

6.4.8 Usage over time

Usage over time is not directly related to the benefits of social navigation, but it can give us valuable information on when Kalas was used and how much it was used. It could give us a clue as to why certain effects cannot be seen. Figure 6.3 through Figure 6.5 display various statistics of when Kalas was used during the six month period. Not surprisingly users are most active between 3pm to 9pm, which is after work hours.

Figure 6.3 shows how many people were logged in throughout the entire evaluation and Figure 6.4 the corresponding activity (measured in viewed recipes). There are a couple of things to note. Apart from a few radical peaks, the amount of users logged in is fairly stable. To us, this is a positive evidence that in general Kalas worked well, and that the reason that people did not chat or did not see others move between recipe collections was simply because there were not enough active users in the system – and not because the system was hard to use in the long run. The second fact to note is that both the number of logged in users and viewed recipes seem to increase over time (the trend is slightly higher for the number of viewed recipes. This can be an indication of learning. When users have familiarized themselves with the system they view and select more recipes.

A community needs activity and when the number of users within a community is under a certain threshold, activity (or content) has to be provided from outside. The peaks in Figure 6.4 and Figure 6.5 arise from "outside interference". The first activity peaks (around day 48 and 70) coincided with two emails sent to the Kalas users; one explaining a potential problem and one informing of an updated version of the system. The next major peak happened when hemma.net sent out an email to all their users (around day 105). The last two peaks (around day 120 and 140) we sent out emails to the Kalas users. Each activity peak is followed with a period of more activity.

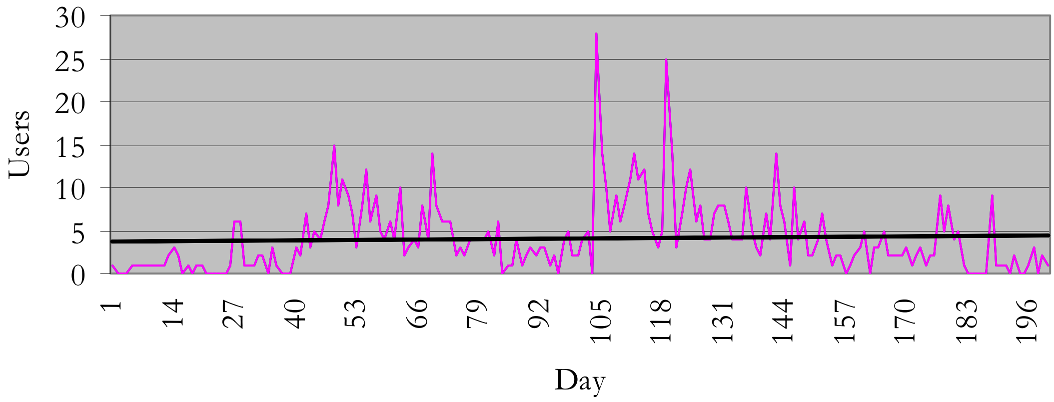

Figure 6.3: Number of users per day. The black line marks the trend over the period.

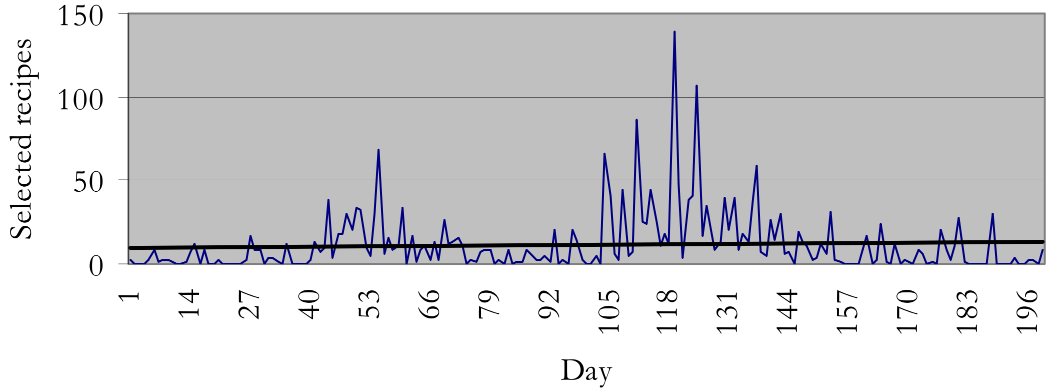

Figure 6.4: Selected recipes per day. The black line marks the trend over the period.

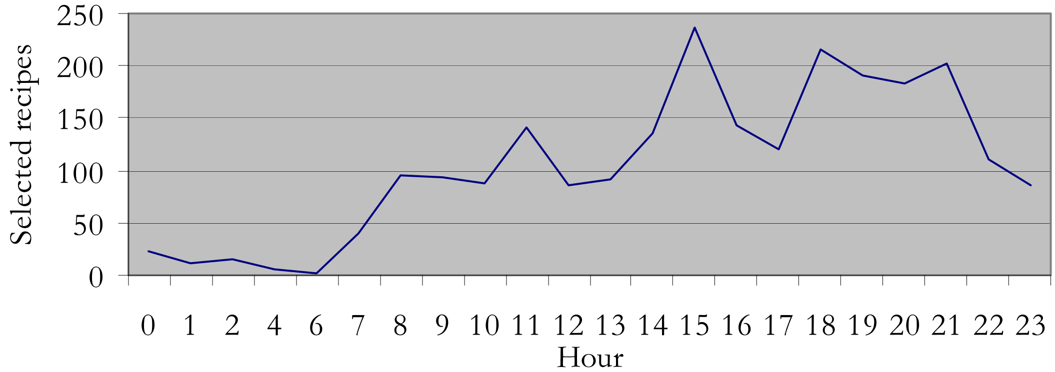

Figure 6.5: Selected recipes during a day.

6.4.9 In-depth interviews

Four users were selected for in-depth interviews. They were selected randomly from those users who had used the system during the last month it was evaluated. The interview attempted to provide a richer understanding of the post-questionnaire replies. In particular, we were interested in:

Task. Why does the user use Kalas, is it to find recipes or to socialize?

Recipe list. When the user is presented with the ranked recipe list she is encouraged to give her view of it, for example, if it reflects her personal taste. She is also asked to sort the recommended list as she would like and comment on each recipe she would like to try and not like to try.

Recipe. What are the factors that are important to our subject when she selects a recipe, and why?

Privacy. Are the users invisible, and if so, what are the reasons for being invisible? Did the subjects feel uncomfortable when others could see when they changed collections? Did they choose not to do certain actions when others were in the system?

Usage reshapes the system. Are the users aware of the fact that their actions affect the system? Did they actively try to influence Kalas and what are the reasons for doing so?

Chatting. Do users chat and for what reason? What do users want to talk about?

People attract people. The user is presented with a picture of a fairly populated overview map and asked which collection she would turn to. If people attract people she will choose one of the more populated collections.

In the presentation we will call the users John, Maria, Anna and Mia.

John: 42 year old male

John works professionally with food – as a chef – and used Kalas to search for recipes. When he used the system he never saw any other users. He would have preferred a more traditional division of recipes in the system, with collections referring to meat, fish, and vegetarian food. Dividing food that way would have been suitable for him, the restaurant where he worked had daily specials with fish, meat, and vegetarian food.

John did not see himself as particularly "social" in the sense that he immediately goes out and talks to strangers. He used his login name at other online sites. The only personal information he gave out was an anonymous email address (i.e. hotmail).

He believed that the recommender system chose the recipes that were the most popular. Thus, he both noticed the feature and reflected over it. He found the recommendations good and wanted to try out the recipes. He said that he is always looking for new recipes and consequently found that the recommender fitted his needs. He did not bother to sort the recipes in any other order due to the reason just given.

In the actual process of deciding on a recipe he looked at the ingredients. The other information was of no use to him. As a chef he was probably able to determine the value of a recipe from just looking at the ingredients.

The only functionality John used in the system was to move between collections and look for recipes. He was always visible and did not feel uncomfortable with other users being able to see his actions. John found Kalas useful mainly due to the number of recipes it contains.

MARIA: 68 YEAR OLD FEMALE

Maria used Kalas to get recipe hints. She found the system quite useful, but stopped using it when she got a virus infected mail from another Kalas user.

Mary was open about herself. She used her real name as login, partly because it was easier to remember, and partly because she did not mind giving it away. She had no problem giving out her full name, email address, and phone number. She did not see the point in being secret about that.

Maria was not too comfortable with the way the system ordered recipes. In particular she did not understand the "thumbs up" symbol. She did however appreciate and saw the point in dividing the space (e.g. recipes) into collections that matched the need of different types of people.

As well as John she looked at the ingredients when deciding if a recipe was worth cooking or not. She claimed not to have used any other functionality other than just looking at recipes. However, Maria said that she read about other people who were in the system when she was and actually got in contact with one via email. Clearly, she did not regard those "social functions" as part of the system functionality. She claimed she would never chat.

The best thing about Kalas was that it offered a new way of finding recipes. The worst thing about Kalas was that it was difficult to install.

When showed a picture of a very crowded Kalas – with many users in one particular collection - Maria claimed that she would go to the collection that most likely contained the recipes she was after.

ANNA: 61 YEAR OLD FEMALE

Anna works with food and used Kalas because she thought it was fun. Anna used her real name as login name but did not give out her email address. She found the age information valuable to provide; in case one would want to chat the age was a good way to find if two people have anything in common. We have already seen that age was the personal information users provided most often and this may the may be the general reason for providing it.

Anna found the recipe list good and, as she explained, used it to search for recipes because of her profession. She chose recipes based on ingredients. Anna wanted to read the comments a recipe had, but felt she was the only one who wrote comments.

Anna liked the division into recipe collections, but felt it lacked some additional filtering into, for example, desserts and main courses.

She had a fairly good idea of how the recommender system worked. She believed it recommended recipes that other users had voted on. As a consequence, she decided to provide explicit feedback and saved recipes in "My Box". She deliberately tried to influence the system to recommend the recipes she finds interesting.

Being visible was no problem to her. In fact, the only person she ever met in the system she immediately tried to make contact with in the chat. She was very interested in making contact with other users in the system that shared her interests.

Mia: 25 year old female

Mia, like the others, used Kalas to search for new interesting recipes. She used her real name as login (both first name and last name). She did not think about the consequences of giving her real name, although she mentioned that she sometimes regrets it afterwards. In most cases when an email address is required, she provides her hotmail address, which is what she did in Kalas.

Like Anna she found the recipe list useful and the recommended recipes interesting. She did not reflect on how Kalas recommended recipes. The division into recipe collections was good and made it easy to find recipes.

Mia was an active person who voted on recipes she cooked. She also used explicit negative feedback, by marking recipes she thought would taste awful with "thumbs down". When choosing a recipe she usually looked at the name and sometimes the recipe picture.

Mia used most functionality that Kalas offered, she saved recipes, made shopping lists, and read comments about recipes. Mia was comfortable about being visible to others. On the other hand, no one was in the system when she was. She did, however, mention that she gave some personal information in case someone else was around – which would have been fun. If she had seen another person in Kalas she would have moved to the appropriate collection to start a discussion in the chat.

The best part of Kalas was the simple interface and how easy it was to find what she needed. She also found the interface appealing and the system fun to use. On the negative side she felt that Kalas was somewhat difficult to use in the beginning. She got lost and had some difficulties to find her way back.

Implications for the overall result

There are a couple of connections we want to make between the interviews and the overall results.

- John, Anna, Maria and Mia confirm that the recommender system was useful. The recommendations were "fun" and triggered their curiosity.
- They actively tried to influence Kalas. This result is different from the outcome of the post-questionnaire. However, they do not see those actions as "deliberately" modifying the space.

- Anna, Maria and Mia wanted to read about others, chat, and read comments. This shows that they do indeed want to be influenced by what others are doing. Thus, the social layer is a valuable tool in the navigational process.
- Giving out personal information is not just done "automatically". It provides additional value. Maria read about the other users and Anna provided her age as a way to find "like minded" users.

6.5 DISCUSSION

In Kalas we have managed to design a system that really makes users aware of the social functionality. Most users found the system easy to use, understood that they could vote on recipes and comment on recipes. Overall the social features were appreciated. We believe both the EFOL study and Kalas study show that social navigation works and adds value. Below we explain why.

DESIGNING FOR SOCIAL NAVIGATION

Both EFOL and Kalas show that it is possible to design for social navigation. The EFOL study showed that the design process is iterative. The one thing that stands out in designing for social navigation is to make the social functionality visible.

Explaining recommendations: In EFOL it was difficult for users to understand the recommendations but when recommendations were explained by the "thumbs up" symbol, users understood and appreciated them. The very strong connection between giving feedback and receiving recommendations (both were indicated with the "thumbs" symbols) made it easier for users to understand that people's choices affected the recommendations. The Kalas study confirms the results found in (Herlocker et al., 2000), recommendations can be explained by simple visualizations of the inner workings of the system.

Delayed feedback: We have to design the system so that users can easily give feedback to it. In Kalas this was partly achieved by allowing users to comment on and vote on recipes more or less anywhere in the system. However, of most importance was that the design allowed users to imminently react to past choices when they entered the system. Most users also gave negative votes on recipes in "My Box", i.e. recipes they had already chosen. This definitely shows that users are willing to give feedback to the system and that Kalas allowed them to do so.

SOCIAL QUALITY ADDS VALUE

The social layer was appreciated; the users felt that the system became richer with it. Since the social features were not necessarily used to find recipes, it shows that the social layer is valuable in its own right. The social layer enhanced the users' experience with the system, although it did not always explicitly affect users' choice of recipes.

Users filled in a profile, gave explicit feedback, wanted to comment on recipes, and chat with others. They wanted to leave their mark in Kalas. The in-depth interviews give clues to why such a small number of users claimed not to have actively influenced the system. The fourth user (Mia above) actively gave negative feedback on recipes she probably would not like but, at the same time, claimed she never tried to influence the system. As in real life social navigation in Kalas happened on a very subtle and unconscious level. Users are influenced by the social layer without them upon it.

RECOMMENDATIONS

Bootstrapping recommendations: Bootstrapping is considered to be one of the more difficult problems to overcome in collaborative filtering and from a pure performance perspective this is probably true. By performance we mean the ability to predict users' rating of information. What we want to argue is that bootstrapping is not a problem if we look at a recommender system from a slightly different perspective. The study gave strong indications that the number of recipes chosen from recommendations is fairly constant over time.

Recommendations are not a replacement for other types of searching but used as a tool to get inspired. In the same way as the social quality adds value, the recommendations add a dimension to the recipe list. By putting a "thumbs up" symbol next to a recipe we trigger a user's curiosity and the system feels more interesting to use. The difference between Kalas and other recommender systems is that users do not have to make a number of initial selections before receiving recommendations. It is when users have to put in a lot of work to get the recommender system going that bootstrapping becomes a problem. When it is seen as a tool that is well integrated with the rest of the system (just marking some recipes based on some analysis of user behavior), it is not that important that recommendations are perfect.

Evaluating recommendations: The previous point inevitably leads to the next point, namely how to evaluate a recommender system. A traditional analysis of the recommender would not have showed that bootstrapping is less of problem than expected. Looking at pure algorithmic properties the system (e.g. precision and recall) will not accurately measure the performance of a recommender system. Instead, we have to focus on the *usefulness* in terms of how and why users act on recommendations. Both the subjective views and usage analysis suggest that people like recommendations. As with the other social features recommendations also inspire people and help them explore the space. Precision and recall will not evaluate the purpose of a recommender system.

SOCIAL NAVIGATION IS NOT A REPLACEMENT

Social navigation is not a replacement for other navigational aids it is a complement. We can focus too much on trying to prove that social navigation works better than other types of navigational aids, when it in fact offers an additional entrance to a space.

The in-depth interviews showed that the recipe collections served as good filters and that users even wanted more ways to access information. We believe that the recommender system worked well because of the recipe collections. Many users first turned to the collection of choice, and then, among all good recipes the recommender pointed out what others like. In Section 5.2 we speculated that social filtering works well under conditions in which a user has several good choices to choose from, i.e. the scenario in which "a lot of information is useful out of which some is recommended". This is what we are seeing in Kalas. Together with the other navigation aids social navigation constitutes a powerful navigation mechanism for people to use.

PRIVACY AWARE BUT WANTS ACCESS TO THE SOCIAL FUNCTIONALITY

Although not many people use the ability to be invisible, still around 17% feel uncomfortable that others can see them or chat with them. This result confirms our previous findings that social navigation support has to be designed with this user group in mind. Interestingly, the privacy aware users in Kalas were voluntary but still completed the entire study. Thus, there are a group of users who are willing to sacrifice some of their privacy to use the system.

What is seen as privacy intrusive is individual. Some feel more uncomfortable with chatting while others feel uncomfortable with being seen by others. In the EFOL study users were more bothered by chatting while in this study users felt more uncomfortable with being seen. We believe that this result is partly due to how the different studies were set up. In EFOL users were not anonymous but in Kalas users were complete strangers.

SOCIAL NAVIGATION HAPPENS UNCONSCIOUSLY

There seems to be a recurrent pattern that a lot of the social trails are created and used without users noticing them. The social features are not necessarily perceived as supporting navigation in terms of getting from one location to another albeit the fact that users wanted to give and read comments, and acted on recommendations. We see that social navigation also helps users to explore a space, makes them visible in space, and influences the space, i.e. turning it into a place. We see a trend where users go to populated collections even though many say they prefer collections with few users. Social affordance and following users around are probably two activities that take place without users realizing it. For example, if someone is asked why she chooses to go to a particular restaurant, she will say that it is because of the food and not because of the number or kinds of people in it. Along the same lines, a person will probably not claim that the reason she is quiet at the library is because everyone else is. These are silent, unconscious processes that happen all the time without us being aware of them. And, in the end, this helps users to find what they are looking for.

Therefore, we believe that social navigation should not be evaluated in the same way as more traditional navigational aids such as maps or signs (which have a very clear purpose). We have seen that social quality is not used in the same way as, for instance, recipe ingredients. The four in-depth interviews showed that users separate recipe searching from other activities. People like to believe that they make objective judgments before choosing something. Three of our in-depth interviewees also stated that it was the ingredients that made them choose a recipe.

6.5.1 Lessons learnt

The in-depth interviews point to one of the main difficulties that can be observed in evaluating social navigation. Many of the effects can only happen if the user population is above a certain threshold – a threshold that is domain dependent and thus hard to determine in advance for a particular domain. Social affordance, synchronous communication and online awareness will only happen if there are a sufficient number of people leaving their traces in the system. As one user pointed out: *it is not worth making comments if you are the only one doing it.* For some of the effects we wanted to evaluate 302 users were not enough. However, all social functionality in Kalas was both visible and appreciated by most users. Given more users and time we would expect, for example, to see more social affordance happening.

Although the Kalas design made the social feature very visible it could have been designed even better. Many users appreciated comments and knew that the functionality was there but did not use it. If the comments had been visible directly in recipes more users would have used them. Thus, the integration (see Section 3.2.5) between recipes and comments could have been clearer.

Social navigation should be designed so that it becomes a part of the overall texture used to determine the value of individual items. In our study we asked users to rate which factors were important in choosing recipes, thus, in effect, they had to disentangle the texture and weigh different factors against one another. It would have been better to find ways of evaluating the entire texture, or, at least, to make the questions differentiate between the connotative attributes (e.g. recipes) and the denotative attributes of recipes (e.g. ingredients) as discussed in Section 2.2.

The evaluation showed that the designer could actively make a system be more or less used. Whenever we sent out community mail we saw an increase in usage. There are studies that show that community based systems need active intervention from its moderators (Girgensohn and Lee, 2002). The system designers have to show that the community is active and live.

7 CONCLUDING REMARKS

When this thesis work started research in social navigation was just in its cradle. Since then it has matured and become an established research field. Today we have a fairly common understanding of social navigation and the different ways that we can design for it. Commercial companies have picked up on some of the ideas presented in here, namely to include various ways for people to communicate and help each other. Today, collaborative filtering is a common feature in most e-commerce web sites.

Let us end this thesis by revisiting the objectives and then discuss some possible opportunities that were disclosed in our work.

DEFINITION

The approach was to narrow the problem down to view social navigation from a purely navigational perspective. Social navigation is a tool for supporting navigational activities. In that respect our view differs slightly from how many others view social navigation.

The reason for viewing social navigation from a navigational perspective is more theoretical than practical. Many of the ideas developed could and should be applied to situations where the prime target is not to navigate a space. It is easy to see how the benefits of social navigation (as defined in here) are applicable to other situations as well. The Kalas evaluation showed that many users do not view social navigation as a navigational aid, which further stresses openness towards what it means to support navigation.

DESIGN

To tackle the problem on how to design for social navigation there are several ways to go. Our approach was to first define a set of design principles and implement those in a designer's toolkit. By acting as the designer we implemented two systems based on our understanding of social navigation. Thus, with our notion of what social navigation is we showed by example that the design principles fitted that framework.

Since we set out to design a system that supported as many ways of social navigation as possible, from direct social navigation to collaborative filtering, the design principle of appropriateness had to be left out. However, the EFOL study showed that many effects and flaws can be caught by simple means. Thus, the way to design for social navigation is to first create a system that supports social navigation in full and then do a quick and dirty evaluation of the prototype.

Evaluation

Does social navigation work and was it possible to take a phenomenon happening in the real world and design for it in electronic systems? Both the EFOL study and Kalas show that people use social navigation in their navigational process. Considering that we had people with very different backgrounds (ranging from computer students to elderly women) this is very promising. In Kalas 92% of the users understood that it was possible to comment recipes and 83% understood that they could vote on recipes.

We believe that users have a tendency to divide a space into different places much in the same way as they do in the real world. Consider the following snippets from the chat sessions in Kalas: "hello"; "I have been kind of lonely in kalas". Now consider the following comments: "The plum pie was very good and easy to cook [...]". Thus, users saw Kalas as consisting of several places with different purposes. The comments were very much targeted towards saying things about the recipes while the chat area was more of a leisure place. The absence of other users, thus, became stronger in the chat area.

Our framework for evaluation allowed us to look at the recommender system in Kalas slightly different than we usually do. Instead of evaluating the systems' capability to predict a user's vote, we focused on how users acted on recommendations. The framework gave us the means to explain why bootstrapping a recommender system is less of a problem than we expected.

Finally, not very many large-scale studies of social navigation have been conducted, which makes it difficult to draw conclusions for the general case. What the thesis partly shows is the need for more evaluations of social navigation systems. There are trends that we want to investigate further. We have provided a framework for how to evaluate social navigation and an in-depth analysis of how social navigation works in the food recipe domain.

7.1 Opportunities for social navigation

The Kalas evaluation showed that a recommender system works well when combined with other kinds of filtering. Content-based and collaborative filtering have similar goals but different approaches. Each approach has its particular problems that may be avoided when the two are combined. For instance, in collaborative filtering, there is no distinction between two items that have the same set of votes or opinions. A simple solution to this problem is to categorize items based on the content. We have started to look at how traditional information retrieval techniques can be applied to collaborative filtering (Cöster and Svensson, 2002).

Secondly, we saw that users want to find other users with similar interest. It is possible to move away from the notion of viewing recommender systems as systems for recommending "things" to systems that recommend people. By visualizing users instead of items we could use a recommender system as a way of creating communities. If a community is defined as "a group of people that share a common interest either implicitly or explicitly", a recommender system would support users in finding community members automatically and help people to find communities that they did not know they were interested in.

FINAL REMARK

While social navigation is interesting, it is even more interesting to broaden the perspective from only looking at navigation. What we want is a shift in the way we perceive and use computers. It is no longer enough to view computers merely as tools. We see social navigation as being part of the larger field of *social computing*. Computers are used as a place to meet friends, socialize, and in general connecting people in various ways. This way of computing calls for a shift in design. Not only should a program be functional; it should also be fun, invoke curiosity, and in general get at the subjective experience of using the program. Social navigation is a step towards the social way of interacting with computers.

8 BIBLIOGRAPHY

Mark S. Ackerman. 1994. "Augmenting the organizational memory: a field study of answer garden". In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 243-252, Chapel Hill, NC, ACM Press.

Mark S. Ackerman and Thomas W. Malone. 1990. "Answer Garden: a tool for growing organizational memory". In *Proceedings of COIS90 – Conference on Office Information Systems*, pp. 31-39, Cambridge, MA, April.

Mark S. Ackerman and Dave W. McDonald. 1996. "Answer Garden 2: merging organizational memory with collective help". In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*, pp. 97-105, Boston, MA, November. ACM Press.

Mark S. Ackerman and Brian Starr. 1995. "Social activity indicators: interface components for CSCW systems". In *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pp. 159-168, Pittsburgh, PA, November. ACM Press.

Irina Akoulchina and Jean-Gabriel Ganascia. 1997. "SATELIT-Agent: An Adaptive Interface Based on Learning Interface Agents Technology". In *Proceedings of the Sixth International Conference on User Modeling*, pp. 21-32, Sardinia, Italy. Springer.

Robert St. Amant and Martin S. Dulberg. 1998. "An experiment with navigation and intelligent assistance". In *Proceedings of the 3rd international conference on Intelligent user interfaces*, pp. 171-178, San Francisco, CA, January. ACM Press.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison-Wesley.

Rob Barrett, Paul P. Maglio, and Daniel C. Kellern. 1997. "How to personalize the web". In *Conference proceedings on Human factors in computing systems*, pp. 75-82, Atalanta, GA, April. ACM Press.

Nicholas J. Belkin and W. Bruce Croft. 1992. "Information filtering and information retrieval:two sides of the same coin? ". *Communications of the ACM*, 35(12), pp. 29-38.

R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor, G. Woetzel. 1997. "Basic Support for Cooperative Work on the World Wide Web". *International Journal of Human-Computer Studies*, 46(6), pp. 827-846.

David Benyon and Kristina Höök. 1997. "Navigation in Information Spaces: supporting the individual". In *Proceedings of the 6th IFIP Conference on Human-Computer Interaction*, pp. 39-46, Sydney, Australia, July. Chapman & Hall.

John S. Breese; David Heckerman; and Carl Kadie. 1998. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", pp. 43-52, In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI, May. Morgan-Kaufman.

Darrell L. Butler, April L. Acquino, Alica A. Hissong, and Pamela A. Scott. 1993. "Wayfinding by Newcomers in a Complex Building". *Human Factors*, 35(11), pp. 159-173.

Elizabeth F. Churchill, Jonathan Trevor, Sara Bly, Les Nelson, and Davor Cubranic. 2000. "Anchored conversations: chatting in the context of a document". In *Proceedings of the CHI 2000 conference on Human factors in computing*, pp. 451-461, The Hague, The Netherlands, April. ACM Press.

Jason A. Collins, Jim E. Greer, Vive S. Kumar, Gordon I. McCalla, Paul Meagher, and Ray Tkatch. 1997. "Inspectable User Models for Just-In-Time Workplace Training". In *Proceedings of the Sixth International Conference on User Modeling*, pp. 327-337, Sardinia, Italy, June. Springer.

Jeff Conklin. 1997. "Hypertext: An Introduction and Survey". *IEEE Computer*, 20(9), pp. 17-41.

Rickard Cöster. 2002 "Learning and Scalability in Personalized Information Retrieval and Filtering". Licentiate Thesis of Philosophy, University of Stockholm, Stockholm, Sweden.

Rickard Cöster and Martin Svensson. 2002. "Inverted file search algorithms for collaborative filtering". In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 246-252, Tampere, Finland. ACM Press.

Andreas Dieberger. 1994. "Navigation in Textual Virtual Environments using a City Metaphor", Doctoral Thesis of Philosophy, Vienna University of Technology, Vienna.

Andreas Dieberger. 1996. "Browsing the WWW by interacting with a textual virtual environment—a framework for experimenting with navigational metaphors", In *Proceedings of the seventh ACM conference on Hypertext*, pp. 170-179, Bethesda, MD. ACM Press.

Andreas Dieberger. 1997. "Supporting Social Navigation on the World-Wide Web". *Internatinal Journal of Human-Computer Studies,* 46(6), pp. 805-826.

Andreas Dieberger. 2002. "Social Connotations of Space in the Design for Virtual Communities and Social Navigation". In K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp. 293-313. Springer.

Andreas Dieberger and Kristina Höök. 1999. "Increasing Awareness of Browsing and Editing Activities in a Virtual Web Community", In *Proceedings of WebNet 99*, 289-294, Honolulu, Hawaii. AACE.

Andreas Dieberger Paul Dourish, Kristina Höök, Paul Resnick, Alan Wexelblat. 2000. "A. Social Navigation: Techniques for building more usable systems", *interactions*, 7(6), pp. 36-45.

Andreas Dieberger, Kristina Höök, Martin Svensson, Peter Löönqvist. 2001. "Social Navigation Research Agenda", In *Extended abstracts of the SIGCHI conference on Human factors in computing systems,* pp. 107-108, Seattle WA, May. ACM Press.

Paul Dourish. 2002. "Where the Footprints Lead: Tracking Down Other Roles for Social Navigation", in K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp. 273-291, Springer.

Paul Dourish and Matthew Chalmers. 1994. "Running Out of Space: Models of Information Navigation", In *Proceedings of Human Computer Interaction*, Glasgow, UK, August.

Roger Downs and David Stea. 1973. "Cognitive Maps and Spatial Behavior: Process and Products", In R. Downs and D. Stea (eds.), *Image and Environment*, pp. 8-26. Aldine.

John Eklund, James Sawers, and Romain Zeilinger. 1999. "Nestor Navigator: A Tool for the Collaborative Construction of Knowledge Through Constructive Navigation", In *Proceedings AUSWEB99*, pp. 396-408, Lismore, Australia, 396-408.

Thomas Erickson. Thomas. 1996. "The World Wide Web as Social Hypertext". *Communications of the ACM,* 39(1), pp. 15-17.

Thomas Erickson, David N. Smith, Wendy A. Kellogg, Mark Laff, John T. Richards, and Erin Bradner. 1999. "Socially translucent systems: social proxies, persistent conversation, and the design of "babble"", In *Proceeding of the CHI 99 conference on Human factors in computing systems: the CHI is the limit*, pp. 72-79, Pittsburgh, PA, May. ACM Press.

Lenord N. Foner. 1997. "A Sociological Analysis of a MUD Chatterbot". In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, February.

Mattias Forsberg, Kristina Höök, and Martin Svensson. 1998. "Design Principals of Social Navigation", In *Proceedings of the 4th ERCIM Workshop on User Interfaces for All (UI4All'98)*, Stockholm, Sweden, October. http://ui4all.ics.forth.gr/UI4ALL-98/forsberg.pdf.

Maria Frostling-Henningsson. 2000. "Dagligvaruhandel över nätet… vad innebär det? En kvalitativ studie av 22 svenska hushåll 1998-1999", Licentiate of Philosophy, School of Business, Stockholm University, Sweden.

Don Gentner and Jakob Nielsen. 1996. "The Anti-Mac Interface". *Communications of the ACM,* 39(8), pp. 70-82.

Andreas Girgensohn and Alsion Lee. 2002. "Making Web Sites Be Places for Social Interaction". In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pp. 136-145, New Orleans, Lousiana. ACM Press.

Frank Halasz and Thomas P. Moran. 1982. "Analogy Considered Harmful", In *Proceedings of the first major conference on Human factors in computers systems*, pp. 383-386, Gaithersburg, MD, March. ACM Press.

R.H.R Harper. 2002. "Information that Counts: A sociological View of Information Navigation". In K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp. 343-353, Springer.

Steve Harrison and Paul Dourish. 1996. "Re-place-ing space: the roles of place and space in collaborative systems". In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pp. 67-76, Boston, Mass. ACM Press.

Jonathan L. Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. "An Algorithmic Framework for Performing Collaborative Filtering", In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, pp. 230-237, CA, USA. ACM Press.

Jonathan Herlocker, Joseph A. Konstan, and John Riedl. 2000. "Explaining Collaborative Filtering Recommendations". In *Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work*, pp. 241-250, Philadelphia, PA, December. ACM Press.

Will C. Hill, James D. Hollan, Dave, Wroblewski, and Tim McCandless, Tim. 1992. "Edit Wear and Read Wear", In *Conference Proceedings on Human Factors in Computing Systems*, pp. 3-9, Monterey, CA, USA, May. ACM Press.

Will C. Hill and Loren Terveen. 1996. "Using frequency-of-mention in public conversations for social filtering". In *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, pp. 106-112, Boston, MA, ACM Press.

Kristina Höök. 1996. "A Glass Box Approach to Adaptive Hypermedia". Doctoral thesis of Philosophy, Stockholm University, Stockholm.

Kristina Höök. 1998. "Evaluating the Utility and Usability of an Adaptive Hypermedia System", *Knowledge-Based Systems,* 10(5), pp. 179-186.

Kristina Höök, Per Persson, and Marie Sjölinder. 2000. "Evaluating Users Experience of a Character Enhanced Information Space". *Journal of AI Communications*, 13(3), pp. 195 212.

Kristina Höök and Martin Svensson. 1999. "Evaluating Adaptive Navigation Support". In A. Munro, K. Höök, and D. Benyon (eds.), *Social Navigation of Information Space*, pp. 240-251. Springer.

Eric Horvitz. 1997. "Agents With Beliefs: Reflections on Bayesian Methods for User Modeling". In *Proceedings of the Sixth International Conference on User Modeling*, pp. 441-442 Sardinia, Italy, June. Springer.

Piero Impicciatore, Chiara Pandolfini, Nicola Casella, and Maurizio Bonati. 1997. "Reliability of Health Information for the Public on the World Wide Web: Systematic Survey of Advice on Managing Fever in Children at Home". *British Medical Journal*, pp.1875-1879, 314(7098).

Jussi Karlgren 1990. "An Algebra for Recommendations". Technical Report 179, Syslab, Department of Computer and System Sciences, Stockholm University, Stockholm, Sweden.

Jussi Karlgren. 2000. "Stylistic Experiments for Information Retrieval". Doctoral Thesis of Philosophy, Stockholm University, Stockholm.

Joseph A Konstan,. Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. "GroupLens: Applying Collaborative Filtering to Usenet News". *Communications of the ACM*, 40(3), pp. 77-87.

Jarmo Laaksolahti. 2002. "Context, Content and Control in an Automatic Collaborative Filtering System". Masters Thesis, Department for Computer and Systems Sciences, University of Stockholm, No. 02-83-DSV-SU, Stockholm Sweden.

Peter Lönnqvist, Andreas Dieberger, Kristina Höök, and Nils Dahlbäck. 2000. "Usability Studies of a Social Enhanced Web Server". In *Workshop on Social Navigation: a design approach? (CHI'2000)*, Hague, Netherlands. http://www.sics.se/-~peterl/chi2000/ Lonnqvist.pdf

Kevin Lynch. 1960. *The Image of The City*, MIT Press

Kevin Lynch. 1973. "Some References to Orientation". In R. Downs and D. Stea (eds.), *Image and Environment*, pp. 300-316, Aldine, Chicago.

Catriona Macaulay. 1998. "Information Navigation in "The Palimpsest"". In *Deliverable 2.1.1 from the PERSONA project*, available from SICS, Stockholm, Sweden.

Paul Maglio and Rob Barrett. 1998. "Adaptive Communities and Web Places". In *Second Workshop on Adaptive Hypertext and Hypermedia (HYPERTEXT'98)*, Pittsburg. http://www.contrib.andrew.cmu.edu/~plb/HT98_workshop/Maglio/Maglio.html

Paul Maglio and Rob Barrett. (1999) "WebPlaces: Adding People to the Web", In *Poster proceedings of the eight International World Wide Web Conference WWW8*, Toronto, Canada.

Paul Maglio and Teenie Matlock. 2002. "The Conceptual Structure of Information Space". In K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp.385-403. Springer.

Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen. 1987. "Intelligent Information-Sharing Systems". *Communications of the ACM*, 30(5), pp. 390-402.

Hannes Marais and Krishna Bharat. 1997. "Supporting Cooperative and Personal Surfing With a Desktop Assistant". In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and* Technology, pp. 129-138, Banff, Canada, October.

Rod McCall and David Benyon. 2002. "Navigation: Within and Beyond the Metaphor in Interface Design and Evaluation". In K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp. 355-384.Springer.

Bradley N. Miller, John Riedl, and Joseph A. Konstan. 1997. "Experiences with GroupLens: Making Usenet useful again", In *Proceedings of the Usenix Winter Technical Conference*, Anaheim, CA, January.

Alan Munro. 1998. "Inhabiting Information Space: Work, Artefacts and New Realities". *Technical report*, (T98:01), Stockholm: SICS.

Alan Munro. 1998b. "Judging Books by Their Covers: Remarks on Information Representation in Real-World Information Spaces". In *VRI workshop Visual Representation and Interpretation*, Liverpool, UK, September.

Alan Munro. 1999. "Fringe Benefits: An Ethnographic Study of Social Navigation at the Edinburgh Festival". In *Deliverable 2.1.1 from the PERSONA project*, available from SICS, Stockholm, Sweden.

Jakob Nielsen. 1992. "Finding Usability Problems Through Heuristic Evaluation". In *Conference Proceedings on Human factors in Computing Systems*, pp. 373-380, Monterey, CA, USA, May. ACM Press.

Blair Nonnecke and Jenny Preece. 2000 "Lurker demographics: counting the silent". In *Proceedings of the CHI 2000 conference on Human factors in computing systems*, pp. 73-80, The Hague, The Netherlands, April. ACM Press.

D. Kevin O'Neill and Louis M. Gomez. 1998. "Sustaining Mentoring Relationships On-line". In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work*, pp. 325-334, Seattle, WA, November. ACM Press.

Leysia Palan Paul Dourish. 2003. "Unpacking "Privacy" for a Networked World". In *Conference Proceedings on Human factors in Computing Systems,* Fort Lauderdale, Florida, April. In press.

Per Persson and Petra Fagerberg. 2002. "*GeoNotes:* a real-use study of a public location-aware community system".*Technical report*, (T2002:27), Stockholm: SICS.

Paul Resnick , Ko Kuwabara , Richard Zeckhauser , Eric Friedman. 2002. "Reputation Systems". *Communications of the ACM*, 43(12), pp. 45-48.

Paul Resnick and Hal R. Varian. 1997. "Recommender Systems". *Communications of the ACM,* 40(3), pp. 56-58.

Alice Richmond. 1996. "Enticing online shoppers to buy - A human behavior study". *Computer Networks and ISDN Systems*, 28, pp. 1469-1480.

Glenna Satalich. 1995. "Navigation And Wayfinding In Virtual Reality: Finding The Proper Tools and Cues to Enhance Navigational Awareness", Unpublished Masters Thesis, University of Washington.

Upendra Shardanand and Pattie Maes. 1995. "Social Information Filtering: Algorithms for Automating "Word of Mouth"", In *Conference Proceedings on Human factors in computing systems*, pp. 210-217, Denver, CO, May. ACM Press.

A.W Siegel and S.H White. 1975. "The Development of Spatial Representations of Large-Scale Environments". In H. W. Reese (ed.), *Advances in Child Development and Behavior*, pp. 9-55. Academic Press, New York.

Sidney Siegel and N. John Castellan, Jr. 1988. *Nonparametric Statistics for the Behavioral Sciences*. McGraw Hill College Div; 2nd edition.

Marie Sjölinder, Kristina Höök, and Lars-Göran Nilsson. 2000. "Age Differences in the Use of an On-Line Grocery Shop - Implications for Design". In *Extended abstracts of Conference Companion of the Human Factors in Computing Systems*, pp. 135-136, The Hauge, The Netherlands. ACM Press.

Marie Sjölinder, Kristina Höök, and Lars-Göran Nilsson. 2003. "Age-related differences in Internet use: The effect of age-related cognitive differences, task complexity and prior Internet experience in the use of an online grocery shop". *Journal of Spatial Cognition and Computation*, forthcoming.

Lynn A Streeter, Diane Vitello, and Susan A. Wonsiewicz. 1985. "How to tell people where to go: comparing navigational aids", *Man-Machine Studies,* 22, pp. 549-562.

Martin Svensson. 1998. "Social Navigation". Technical report (T98:01), Stockholm: SICS.

Martin Svensson, Jarmo Laaksolahti, Kristina Höök, and Annika Waern. 2000. "A Recipe Based On-line Food Store". In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, pp. 260 - 263, New Orleans, Louisiana. ACM Press.

Martin Svensson and Kristina Höök. 2002. "Social Navigation of Food Recipes: Designing Kalas". In K. Höök, D. Benyon, and A. Munro (eds.), *Designing Information Spaces: The Social Navigation Approach*, pp. 201-222. Springer.

Martin Svensson, Kristina Höök, Jarmo Laaksolahti, and Annika Waern. 2001. "Social Navigation of Food Recipes". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 341-348, Seattle, Washington, April. ACM Press.

Loren G Terveen, William C. Hill, Brian Amento, David McDonald, and Josh Creter. 1997. "PHOAKS a system for sharing recommendations". *Communications of the ACM,* 40(3), pp. 59-62.

T. Timpka and N. Hallberg. 1996. "Talking at work - professional advice-seeking at primary healthcare centers". *Scand. J. Prim Health Care 14*, pp. 130-135.

Jonathan Trevor, Thomas Koch, and Gerd Woetzel. 1997. "MetaWeb: Bringing synchronous groupware to the World Wide Web". In *Proceedings of the European Conference on Computer Supported Cooperative Work*, pp. 65-80, Lancaster, UK, 65-80.

Nancy A. Van House, Mark H. Butler, and Lisa R. Schiff. 1998 "Cooperative knowledge work and practices of trust: sharing environmental planning data sets", In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work*, pp. 335-343, Seattle, WA, November. ACM Press.

Alan Wexelblat. 1999. "History-Based Tools for Navigation", In *Proceedings of the 32nd Hawaii International Conference on Systems Sciences (HICSS-32*). IEEE Computer Society Press.

Alan Wexelblat and Pattie Maes. 1999. "Footprints: History-Rich Tools for Information Foraging", In *Proceeding of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*, pp. 270-277, Pittsburgh, PA. ACM Press

Gerhard Widmer and Miroslav Kubat. 1996. "Learning in the presence of concept drift and hidden contexts". In *Journal of Machine Learning*, 21(1), pp. 69-101.

APPENDIX A: KALAS EVALUATION SETUP

PRE QUESTIONNAIRE LAYOUT

Question	Possible answers
Email	<Text>
Zip Code	<Text>
Age	<19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80<
Gender	Male, Female
Are you willing to participate in an in-depth interview of the system you are about to try?	Yes, No, Maybe
Have you any education in or work in any of the following areas (all that applies)?	Construction and maintenance, computers and IT, economics/administration, sales, arts/media, PR/advertising, health care, corporate management, teaching, restaurants, technology, other
Where do you live?	Country side, small town, middle sized town, suburb, major city
How many are there in your household?	1, 2, 3, 4, 5, 6, 7, 8, >8
What are your cooking skills (4 = average)?	Beginners level (1) skilled (7)
How many times a week do you cook?	1, 2, 3, 4, 5, 6, 7, 8, >8
How often do you use computers?	less than 1/month, 1-2/month, 1/week, 2-4/week, every day
How often do you use the Internet?	less than 1/month, 1-2/month, 1/week, 2-4/week, every day
Have you ever chatted on the Internet?	Never, less than 1/month, 1-2/month, 1/week 2 4/week every day
How do you act?	
a. I make contact with people I do not know	Never happens (1) happens often (7)

b. I ask for the way when I am lost	Never happens (1) happens often (7)
c. I share my experiences with others	Never happens (1) happens often (7)
d. When I am on holiday I try to make new contacts	Never happens (1) happens often (7)
Do you use hemma.net for finding recipes?	Never, 1/year, 2-4/year, every other month, 1/month, 2-3/month, 1/week, 2-4/week, every day
If you have used hemma.net what functionality do you use (all that you have used)?	Search on ingredients or categories, Specified search on ingredients, weekly menu, weekend recipes, today's special, send in recipes, advanced search, read articles
Have you ever used another system (apart from hemma.net) to find recipes?	Never, >1/month, 1-2/month, 1/week, 2-4/week, every day

PRE QUESTIONNAIRE CODING TEXT TO NUMBERS

every day = 11

2-4/week = 10

1/week = 9

2-3/month = 8

1-2/month = 7

Once a month = 6

Less than once a month = 5

Every other month = 4

2-4/year = 3

1/year = 2

Never = 1

Post questionnaire

Question	Answer
Is  a symbol for another user?	Yes or no
Could you chat in the system?	Yes or no
Could you comment recipes?	Yes or no
Could you vote on recipes?	Yes or no
How was Kalas to learn? (4 = not easy or difficult)	1 (very difficult) to 7 (very easy)
How was it to use Kalas? (4 = not easy or difficult)	1 (very difficult) to 7 (very easy)
What is your perception of the recipes in Kalas?	
a. (4 = enough recipes)	1 (too few recipes) to 7 (too many recipes)
b. (4 = the recipes were ok)	1 (boring recipes) to 7 (exciting recipes)
c. (4 = half the recipe were useful)	1 (useless) to 7 (useful)
How did the following factors affect your choice of recipe?	
a. The chef/author.	1 (no importance) to 7 (major importance)
b. The level of fat.	1 (no importance) to 7 (major importance)
c. The number of users who had chosen the recipe.	1 (no importance) to 7 (major importance)
d. The recipe's comments.	1 (no importance) to 7 (major importance)
e. The ingredients.	1 (no importance) to 7 (major importance)
f. That it seemed easy to cook.	1 (no importance) to 7 (major importance)
g. That other recommended it in the chat.	1 (no importance) to 7 (major importance)
h The recipe picture of the recipe.	1 (no importance) to 7 (major importance)
i. The recipe's name.	1 (no importance) to 7 (major importance)
j. The cooking time.	1 (no importance) to 7 (major importance)

k. The price range.	1 (no importance) to 7 (major importance)
l. The allergic information.	1 (no importance) to 7 (major importance)
m. From where the recipe was collected.	1 (no importance) to 7 (major importance)
How well do the following statements apply to you?	
a. I prefer recipe collections with many users.	1 (disagree) to 7 (agree completely)
b. I prefer recipe collections with few users.	1 (disagree) to 7 (agree completely)
c. I would choose a collection where there are interesting users.	1 (disagree) to 7 (agree completely)
d. I go to the collections that I believe contains the recipes I am looking for.	1 (disagree) to 7 (agree completely)
e. I learnt something from the others that used Kalas.	1 (disagree) to 7 (agree completely)
f. I deliberately tried to influence the system, for example, by choosing the same recipe over and over again.	1 (never happened) to 7 (happened often)
g. I avoided doing some things, for instance, choose a recipe, since I did not want to influence the system.	1 (never happened) to 7 (happened often)
How do you feel about the fact that others can contact you in the chat? (4 = to some extent uncomfortable)	1 (uncomfortable) to 7 (comfortable)
How do you feel about the fact that others can see you movements? (4 = to some extent uncomfortable)	1 (uncomfortable) to 7 (comfortable)
What do you think about the possibility to chat?	1 (better without) to 7 (worse without)
What do you think about the possibility to comment recipes?	1 (better without) to 7 (worse without)
What do you think about the possibility to vote on recipes?	1 (better without) to 7 (worse without)
What do you think about the possibility to see others)?	1 (better without) to 7 (worse without)
It flickers on the screen when other users	1 (better without) to 7 (worse without)

change collection, how do you feel about that?	
Did you notice that the recipes in the recipe list changed place from time to time when you did not choose to sort recipes in alphabetical order?	Yes or no
If you answered yes on the previous question, did it disturb you?	1 (very disturbing) to 7 (Not disturbing at all)
How does Kalas chooses recipes for you? Only one answer possible	
I have not given it a thought	Yes
Kalas randomly selects recipes to me	Yes
Kalas recommends recipes based on ingredients	Yes
Kalas recommends recipes based on other users' choices	Yes
Kalas recommends recipes based on mine and other users' choices	Yes
Kalas recommends the most popular recipes	Yes
What do you think about the recommend recipes: recipes marked with 	1 (not good) to 7 (very good)
If you have any additional opinions about Kalas, write them here	<text>

IN-DEPTH INTERVIEWS SETUP

Mail was sent to the subjects containing (a) printouts of the recommended recipes they would get in the "all recipes" collection; (b) printouts of the system with many online users; and (c) a printout of the system with the recommended recipe list in the "all recipes" collection.

The interview was conducted over the phone and subjects were informed that the interview was recorded.

One recipe was selected when the subjects explained what factors are important when choosing a recipe.

Printouts were numbered so they could refer to them.

Log data: recorded actions

User created

Login user

Logout user

User changed location

User turned invisible

User updated her personal information

User chatted

User commented

User saved a recipe

User printed a recipe

User saved a recipe to her shopping list

User did a search

User made explicit/implicit votes

Log data: stored action layout

<time><logger><login name><login id><location><user mode><action>

SWEDISH INSTITUTE OF COMPUTER SCIENCE
SICS DISSERTATION SERIES

01: Bogumil Hausman, Pruning and Speculative Work in OR-Parallel PROLOG, 1990.

02: Mats Carlsson, Design and Implementation of an OR-Parallel Prolog Engine, 1990.

03: Nabiel A. Elshiewy, Robust Coordinated Reactive Computing in SANDRA, 1990.

04: Dan Sahlin, An Automatic Partial Evaluator for Full Prolog, 1991.

05: Hans A. Hansson, Time and Probability in Formal Design of Distributed Systems, 1991.

06: Peter Sjödin, From LOTOS Specifications to Distributed Implementations, 1991.

07: Roland Karlsson, A High Performance OR-parallel Prolog System, 1992.

08: Erik Hagersten, Toward Scalable Cache Only Memory Architectures, 1992.

09: Lars-Henrik Eriksson, Finitary Partial Inductive Definitions and General Logic, 1993.

10: Mats Björkman, Architectures for High Performance Communication, 1993.

11: Stephen Pink, Measurement, Implementation, and Optimization of Internet Protocols, 1993.

12: Martin Aronsson, GCLA. The Design, Use, and Implementation of a Program Development System, 1993.

13: Christer Samuelsson, Fast Natural-Language Parsing Using Explanation-Based Learning, 1994.

14: Sverker Jansson, AKL - - A Multiparadigm Programming Language, 1994.

15: Fredrik Orava, On the Formal Analysis of Telecommunication Protocols, 1994.

16: Torbjörn Keisu, Tree Constraints, 1994.

17: Olof Hagsand, Computer and Communication Support for Interactive Distributed Applications, 1995.

18: Björn Carlsson, Compiling and Executing Finite Domain Constraints, 1995.

19: Per Kreuger, Computational Issues in Calculi of Partial Inductive Definitions, 1995.

20: Annika Waern, Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction, 1996.

21: Björn Gambäck, Processing Swedish Sentences: A Unification-Based Grammar and Some Applications.

June 1997.

22: Klas Orsvärn, Knowledge Modelling with Libraries of Task Decomposition Methods, 1996.

23: Kia Höök, A Glass Box Approach to Adaptive Hypermedia, 1996.

24: Bengt Ahlgren, Improving Computer Communication Performance by Reducing Memory Bandwidth Consumption, 1997.

25: Johan Montelius, Exploiting Fine-grain Parallelism in Concurrent Constraint Languages, May, 1997.

26: Jussi Karlgren, Stylistic experiments in information retrieval, 2000

27: Ashley Saulsbury, Attacking Latency Bottlenecks in Distributed Shared Memory Systems, 1999.

28: Kristian Simsarian, Toward Human Robot Collaboration, 2000.

29: Lars-åke Fredlund, A Framework for Reasoning about Erlang Code, 2001.

30: Thiemo Voigt, Architectures for Service Differentiation in Overloaded Internet Servers, 2002.

31: Fredrik Espinoza, Individual Service Provisioning, 2003.

32: Lars Rasmusson, Network capacity sharing with QoS as a financial derivative pricing problem: algorithms and network design, 2002.

33: Martin Svensson, Defining, Designing and Evaluating Social Navigation, 2003.