

RUNNING HEAD

Resource Discovery Assistants

JOURNAL TITLE

International Journal of Human-Computer Studies – Special Issue on Awareness and the WWW

ARTICLE TITLE

Supporting Online Resource Discovery in the Context of Ongoing Tasks with Proactive Assistants

CONTACT AUTHOR

Jay Budzik

Intelligent Information Laboratory

Department of Computer Science

Northwestern University

1890 Maple Ave.

Evanston, IL 60201 USA

+1 847 467-1771 (phone)

+1 847 491-5258 (fax)

budzik@infolab.nwu.edu

AUTHORS

Jay Budzik

Shannon Bradshaw (same affiliation and address)

Xiaobin Fu (same affiliation and address)

Kristian J. Hammond (same affiliation and address)

ABSTRACT

We present ongoing work on systems aimed at improving a user's awareness of resources available to them on the Internet and in intranets. First, we briefly describe Watson, a system that proactively retrieves documents from online repositories that are potentially useful in the context of a task, allowing the user to quickly become aware of document resources available in online information repositories. Next, we describe I2I, an extension of Watson that builds communities of practice on the fly, based on the work that its users do, so that users with similar goals and interests can discover each other and communicate both synchronously and asynchronously. Both Watson and I2I operate given some knowledge of the user's current task, gleaned automatically from the behavior of users in software tools. As a result the systems can provide users with useful resources in the context of work they are performing. We argue the systems can foster a greater sense of awareness of the resources available, while minimizing the effort required to discover them.

KEYWORDS

resource awareness, proactive assistants, information agents, just-in-time retrieval, agent-application integration, user interfaces, context

PAGES

30

1. Introduction

The Web is an incredible resource. It has put a wealth of information and diverse communities of people within the reach of millions around the world. Yet while users go about their daily tasks on a computer, either on the Web or in more traditional applications, they are typically unaware of the potentially useful resources available in this new, highly connected environment. This means that even though many potentially useful resources are available to them, they may not take advantage of these resources because they don't know they are there, or because the effort required to access them outweighs the benefit they expect to accrue. Users must first perceive the need to become aware of available resources, decide to find them, and, finally, they must know how to use available tools effectively to do so.

Our view is that the current level of effort required to leverage existing resources is problematic. Our work aims squarely at reducing the friction typically associated with discovering potentially useful resources on the Internet and in intranets. Our approach is to provide users with software assistants that can proactively recommend relevant resources in the context of the task a user is performing by observing the user work in existing software applications.

In this article, we describe systems that can (1) proactively retrieve relevant documents from online repositories and (2) put users in touch with others who may be helpful collaborators in the context of the work in which they are currently engaged. We provide evidence that suggests the software can benefit users by making them more aware of useful resources available to them.

In order to support the awareness of relevant resources in the context of the tasks users are performing, and without requiring additional overhead from the user, the systems we describe must be coupled closely with tools commonly used to perform these tasks. This coupling allows the systems themselves to become aware of what the user is doing, and then react to the user by proactively providing her with access to useful resources. Watson and I2I, two of the systems we describe, are closely coupled with existing document manipulation applications, and attempt to make users aware of relevant documents and people by constructing and manipulating lexical representations of the user's current work product. These representations are used to access networked resources on behalf of the user, typically by querying information retrieval systems.

Consider the following examples of how someone might use these systems. These examples have all been worked out using current versions of the implemented systems.

1.1 Usage scenarios

1.1.1 Watson

Watson attempts allow the user to easily maintaining an awareness of written resources available to her in the context of an ongoing task, such as writing a paper. It provides references to additional information gathered from online information repositories as a user works using common applications like word processors and Web browsers (see Figure 1).

Suppose, for example, our user is a student doing research on the relationship between Taiwan and China. She received an email from her teacher with her assignment that included a link from the Chinese Embassy, stating China's position on reunification. She notes this document's extreme bias on the side of reunification, and would like to find other documents that take a different perspective. The Chinese Embassy pages are of no help to her; they have no links to external sources. She goes to her Watson window. Watson has been watching her browse, and has automatically gathered a list of related documents. One particular document catches her eye: a policy article from the US embassy about Taiwan and China. This document has a distinctly different perspective—it is more concerned with the economic impact of a possible reunification for the United States. Watson has also retrieved a news article containing an opinion poll that describes how people in Taiwan are viewing this issue. She opens this link, reviews the statistics and wonders whether or not they can be trusted. She brings up her Watson window. Watson has retrieved documents about the polling organization that supports its integrity, and also an article from the Chinese Embassy criticizing the method used. Effectively, Watson has allowed the user to become aware of the written resources available to her in the context of her current task, while requiring minimal effort on her part.

With current information access tools, a user attempting to perform research like in the example above would have had to engage in a long and tedious process that requires detailed knowledge of information systems and how to access them. After first identifying a need to find additional information, she would have to identify which online information sources could potentially contain documents of interest. For each source she identified, she would have to know how to access it. Then she would have to distill her need into a query to the system. After viewing the results returned, she would probably need to

refine her query a number of times before arriving at a result set that matched her original need. And then she would have repeat this process the rest of the information resources she identified. Our goal is for Watson to automate most of this process, thereby eliminating the need for the user to know (1) what information resources are applicable, (2), how to access them, and (3) how to formulate her needs as a query to those sources.

1.1.2 I2I

I2I is an extension to Watson that attempts to address the problem of maintaining an awareness of the people who could serve as potential resources in the context of the user's current task. I2I puts people who share common work contexts (as represented by the documents they manipulate) in contact with each other (see Figure 2).

1.1.2.1 Scenario One: Expanding opportunities for collaboration across organizational and geographical boundaries

For example, say Mary and Joe are both high school students writing term papers about the environmental impact of pesticides. Joe is from a small town in rural America, and is the only environmental activist in his school. Mary lives in New York and is the president of her school's Earth First chapter, which, among other things, promotes consumer awareness of the benefits of organic farming. Mary and Joe are both using I2I, and as such, the system notices they are writing papers on similar subjects and displays their screen names in a window associated with their current document. Mary and Joe can now contact each other through I2I using text messaging or videoconferencing. The system provides them with an awareness of their shared work contexts and interests, which serves as common ground for the conversation they start about their papers.

1.1.2.2 Scenario Two: Locating expertise and enabling collaboration within an organization

I2I can also be deployed within organizations so they can better leverage their own intellectual capital. Documents contain knowledge of best practices and past work, but often information is more easily obtained from people, and published documents only contain a fraction of an organization's knowledge. Suppose Joe is a consultant who has just finished a proposal to a potential client in the wireless industry for a customer relationship management (CRM) strategy engagement. Joe doesn't want to publish this document, because he isn't sure whether or not it will result in a successful bid. But in the process of doing the work, he learned a lot about the industry and potential CRM implementations that he thinks could be valuable to others who are making similar proposals. He pulls up his I2I window and leaves a calling card in the context associated with his proposal. He says he is available to talk with others who are working on proposals in CRM. Weeks later, Fred, another consultant, has just received an email briefing him about his next engagement, also having to do with CRM. Working off of this email, I2I automatically brings up Joe's calling card because it was left in a similar context to the one Fred is working in now. Fred contacts Joe through I2I, over text messaging. Joe explains to Fred what he knows about CRM and what happened during his proposal process, allowing Fred to benefit from his experiences.

I2I can give users a sense of the people available to talk within the context of an ongoing task by brokering early stages of communication around shared work contexts. I2I can also facilitate capturing and disseminating expertise. With current communication tools, the people in the above scenarios would have either had to know each other beforehand, either by meeting in the physical world, by meeting online through a newsgroup, chat room, or other virtual meeting place. Even if they knew each other, they would have to have detailed knowledge of each other's past and present experiences and interests, in order to become aware that they could potentially help each other. Instead, I2I allows the work people do to serve as the basis for introduction, regardless of the locations they inhabit in physical or virtual space. I2I allows users to publish their interests as they work by leaving calling cards in the contexts established by their work. With I2I, people with relevant experience are more readily available.

1.2 General Relationship to Previous Work

The bulk of previous work in the space of systems for awareness of online resources has focused on supporting continued awareness of previously identified resources across time. For example, Dourish and Bly [12] have investigated the role of various media in supporting a kind of general awareness among coworkers that results in a sense of community. In the context of documents, work has focused on allowing users to become aware of when Web sites change (e.g., [8]).

The work we report here has resulted in tools that provide users with an awareness of potentially useful resources they have not identified beforehand. Users simply go about their work using common software tools, the software tracks their progress and, as a result, it can provide them with potentially useful resources.

1.3 Overview

We go on to describe approaches for enhancing existing tools with awareness information in the context of the task. In particular we describe an architecture that provides useful abstractions to system designers that need to leverage existing applications without modifying them directly. Next, we discuss interface techniques for presenting varying levels of awareness cues that are visually associated with the on-screen representations of the user's current work product. We continue by describing the implementation of Watson and I2I, and close by presenting evaluations of their effectiveness and analysis of their strengths and weaknesses.

2. Architectures for Awareness: Tracking and Representing User Activity in Software Tools

In order to build systems that react to a user as she is working with software tools, the systems must know about the user's interaction with those tools. The software we describe interprets user actions in software applications, and analyzes the product of her work in order to gain enough knowledge of what she is doing to act on her behalf. Generally, two approaches exist for doing this: modifying the software tools themselves, or augmenting the software through operating system or application-specific programming interfaces. By modifying the application to provide awareness facilities (or by designing it with awareness in mind to begin with), the system architect has more control over the entire process. Unfortunately, source code is not always available¹, and so architects must use indirect programming interfaces, either provided by the operating system, or by the application, or both. This approach is taken in the design of Watson and I2I, described in Sections 4 and 5.

2.1 Coupling Awareness Agents with Existing Applications: The Information Management Assistant Architecture

We have developed a reusable architecture to support the coupling of information sources (e.g., Internet search engines, internal knowledge management systems, etc.) with existing applications (e.g., Microsoft Word, Netscape Navigator, etc.) in service of resource awareness. Software "glue" that allows an agent to observe a single application typically does not transfer to other applications. It is important, if the goal is to observe *multiple* applications, to abstract away from the specifics of observing a given application so that effort is not duplicated. The architecture we developed uses the adapter design pattern [15] to achieve this. The Information Management Assistant (IMA) architecture (described in more detail in [5]) provides useful abstraction constructs that allow system designers to add, update, and remove applications and information sources from the system relatively easily. The flow of messages in this architecture is depicted graphically in Figure 3.

Each application and information source is encapsulated in an adapter. Adapters are akin to software plug-ins, in that both add additional functionality using a predefined interface. Systems built using the IMA architecture typically have several *application adapters*, which are used to gain access to an application's internal representation of a document, and the application-level events generated by user interactions with the application. We have developed adapters for Microsoft Word, a word processing package, Microsoft Internet Explorer, Mozilla, and Netscape Navigator, all three Web browsers, Microsoft Outlook, an email client, and Microsoft Power Point, a presentation package. We are currently working on an adapter for Lotus Notes, an email and collaboration system.

Application adapters interpret application-level events so they can be translated into an event representation the agent can process. For example, when user types text into a document in Microsoft Word, keyboard events are generated in the application. The Word application adapter interprets these events, paying mind to their target (in this case, the document the user is modifying). The adapter can then relay a message to the agent, in this case, indicating the document has changed. The agent could then query the application adapter, requesting a representation of the updated document. The adapter would then produce a document representation, which would be sent to the agent for analysis.

It is important to note that under this architecture, adapters can be written in any language. This is particularly beneficial for wrapping applications that have scripting functionality. In this case, adapters can be written in the scripting language of the application, making it possible for Watson to be aware of an application that has no API accessible from an external application, but is customizable from its own internal scripting language.

Based on an analysis of the content of the document the user is manipulating or based on the actions she performs, the agent could decide to query one or many online repositories. The queries the system produces take the form of an internal query representation capable of representing boolean combinations of terms or quoted phrases. This internal representation is sent to selected *information adapters*. Each information adapter translates the query into the source-specific query language, and executes a search. Information adapters are also responsible for parsing the results of a search into a standard representation, much like in metasearch applications [34].

There are many approaches to agent-application integration, some of which can be found in, e.g., [25]. Our contribution in the above architecture is the addition of a layer of abstraction from which various applications and information systems can be treated similarly from the perspective of an agent. This architecture has been useful in building many prototype systems that observe behavior in applications and provide users with information from networked repositories, because the system designers need not focus on the particulars of how an application passes events or to send queries to an information source; instead, they can focus on the core functionality of the application.

3. Interfaces for Awareness in the Context of Ongoing Tasks

The systems we describe are designed to run as a user works, providing assistance along the way. When a user would like to explore available resources in the context of the work she is performing, she can consult the system to see if an item that matches her needs has already been retrieved. There are several pertinent problems in designing interfaces of systems aimed at supporting this kind of awareness in the context of ongoing tasks. First, the interface must balance distracting the user from her primary task with providing her useful information that could merit interruption. Second, the design should allow the user to transparently associate information the system provides with the work it attempts to augment.

Given a chance to design an application from scratch, awareness facilities can be built in directly into the application. However, the systems we describe integrate with existing applications whose source code is not available. As such, we have developed a reusable framework for integrating an agent's user interface with existing applications by attaching graphical elements to the application.

3.2 Interface Attachments for Awareness

When the application cannot be modified to allow for the presentation of awareness cues, the above problems—mitigating distraction and representing correspondence—become more complicated to solve because of the technical constraints on the feasibility of designs. For one, existing application typically do not allow developers to affect anything but limited changes in the application's interface through a provided programming interface (API). For example, it is common to allow developers to add a button or a menu item through programming interfaces, but not much else. Buttons and menu items are useful and versatile graphical elements, but it is difficult to display very much information in either of these elements without breaking established norms.

To address this, we have developed an application-independent interface component for proactive awareness applications (in the tradition of [1]). The attached interface we have developed (we use attachments, here, as in, e.g., [26]) can be displayed in two states: minimized and normal (see Figure 4). In minimized mode, the attachment provides users with graphical representations of the agent's state. In normal mode, the minimized representation is displayed along with the primary interface to the agent. Figure 4 shows the attached interface instantiated for the Watson system.

The attachment follows the window when it is moved and resized. This arrangement allows the user to easily maintain correspondence between their activity and the awareness cues the system presents them with. In addition, the minimized mode allows the user to maintain a peripheral awareness of the agent's state as she engages in her primary task. Users can click on the attachment to display more detailed information in the context of a given window, which is displayed in a custom, agent-specific interface that swings out from behind the application. This design follows Rhodes' notion of ramping interfaces [30]. By arranging interfaces to secondary information items in stages, the interface we built provides a way of managing the tradeoff between the goal to present potentially useful information and the amount the agent distracts the user from her primary task.

The attachment we have implemented contains two graphical interface items that allow the user to quickly determine the agent's state. The first is a traffic light representation, which displays green when the system is finished collecting and verifying results, yellow, when it is in the midst of this process, and red when an error has occurred. The second is a linear representation of the relevance of the items it has gathered (as determined by simple comparison of the query it has generated and the titles and extracts of documents returned by a particular information source). This is conveyed using a row of colored dots. More dots light up when more relevant resources are retrieved.

The attachment interface we have built is application-independent because it is implemented by intercepting the windowing events at the operating system level and processing them to make it appear as the attachment is "glued" to the application window². From an engineering perspective this is important because it allows us to easily allow us to build awareness functionality into a variety of applications, even if they do not directly support interface extensions.

The kind of interface we have described above is important to the usability of any agent that provides information corresponding to existing interfaces. First, it makes the association of information the system gathers with the document they correspond with a visual task. This means determining which results go with which document easier. Second, it allows the user to manage her attentive resources by minimizing the agent's window, while maintaining a peripheral awareness of the system's status.

4. Watson: Supporting Awareness of Relevant Information Resources

Watson [5] provides additional information to users as they are using common applications to write, or to browse the Web (see Figure 1). Watson retrieves relevant information items from Internet search engines and other repositories by analyzing documents users are manipulating in common applications. In order for Watson to respond to user actions and analyze the content of a user's document, it needs to be aware of those actions, and it needs to be able to access the document she is manipulating.

Watson is an instantiation of the IMA architecture described above. Each application adapter is responsible for communicating with a given application. The adapters translate events that occur in the application into an abstract event vocabulary, which it uses to communicate with Watson. Documents in an application are also translated into an abstract document representation and communicated similarly. Likewise, the information repositories Watson uses are encapsulated in information adapters.

Watson analyzes the text of a user's document when it is modified or loaded into an application in order to automatically form queries to online information repositories. The goal of the queries Watson produces is to automatically retrieve potentially useful online information resources, thereby fostering an awareness of the resources available to users. Information resources are gathered automatically from distributed repositories, and then they are clustered using several heuristic result similarity metrics, effectively eliminating redundant results (due to mirrors, multiple equivalent DNS host names, etc.).

Watson's content analysis and query generation algorithms (described in [5]) leverage presentation attributes of the document text (e.g., type style and alignment to name a few) and the frequency of content-bearing words in order to automatically construct a relatively long query to various information repositories. Each application has its own internal representation of a document. For example, Microsoft Internet Explorer represents HTML documents as a tree of tagged nodes (each HTML tag constitutes a new node). Microsoft Word, on the other hand, makes a list of words available, each of which has a different presentation attributes. Application adapters translate this representation into a more general, application-agnostic representation when communicating with Watson. Watson then performs its analysis on this intermediate representation.

Recent versions of Watson have been extended to present information from several different kinds of information sources. Watson can search for related documents from the Internet and intranets (provided an adapter for an indexed repository), images and video from multimedia libraries, items for sale at ecommerce sites, news articles from Internet news sites, and articles from research databases like ProQuest. Instead of presenting the user with a long list of items (Watson typically returns over 50), a summary of the results is presented, and then the actual search results are presented in categories corresponding to their type, in order to avoid overwhelming the user. Each kind of document is associated with a tab. Tabs aligned across the horizontal axis represent major content categories; tabs aligned along the vertical axis represent system categories, including the summary pane. Figure 1 demonstrates this interface. Each part of the summary display shows the type of resource (e.g., News), the number of resources, and an example entry. The user can then click on the corresponding tab (displayed along the top of the interface) to view the rest of the results of that type. Items with associated images (movies, images, and items for sale) are displayed as thumbnails in a grid, along with their caption and URL (see Figure 5). Documents are displayed with title, the text extract returned by the search engine the URL, and the information source or sources that returned the item.

In addition, the Watson window can be minimized (as described in section 3.1) so the user can maintain a more peripheral awareness of the status of the system and the quality of the results retrieved. The value displayed on the quality indicator is proportional to the percentage of documents whose extracts closely match the outgoing query (see Figure 4).

Watson can also process explicit requests for information in the context of the document the user is currently manipulating. When a user submits a query directly to Watson, it combines the new query terms with the previously constructed contextual query by concatenating them to form a single query. In this way, Watson brings the previously gathered information about the context of the user's work to bear directly on the process of servicing a user's explicit query. For example, if a user is viewing a page about NASA's latest Mars mission and enters the query "life", Watson will return a list of pages about life on Mars, not the magazine, the game, the algorithm, nor the biological definition (as several search engines do). Because

Watson grounds explicit queries in the context of the current document, the results returned are coherent, even for this highly ambiguous query.

4.1 Evaluations

We have performed two studies of Watson in order to evaluate whether it can add value to a user's experience by promoting awareness of relevant online documents.

The first study (reported in detail in [5]) evaluated the relevance of Watson's suggestions. We asked users to choose a page from a list containing their bookmarks, look at it in a Web browser and then use Alta Vista to find related pages. The users then judged the top 10 pages returned as relevant or irrelevant to their search task. Next, the users were asked to judge the sites Watson returned from the same page in the same way. In this experiment, Watson used only Alta Vista as well. For our initial group of subjects, we drew from local computer science graduate students. All of the volunteers considered themselves expert-level searchers. This was evident in their query behavior, as most of them used long queries (≥ 4 words), laden with advanced features. We gathered 19 samples from a pool of 6 users. Using Alta Vista, our group of expert searchers was able to pose queries that returned, on average, 3 relevant documents out of 10. Watson was able to do considerably better at the same task, returning, on average, 5 relevant documents out of 10. In the samples gathered, Watson was able to do as well or better than an expert user 15 out of 19 times.

The second study (reported in detail in [6]) evaluated the utility of Watson's suggestions in the context of a task. For this experiment, we asked researchers to submit their last paper and evaluate the results Watson returned on the basis of whether they would be useful during the later stages of their writing process. For the 6 researchers who participated, Watson was able to consistently provide useful information to users in this context, about 2 documents out of 10 using only its automated query construction facilities (not using the query in context functionality at all). One user in this study went so far as to say he was embarrassed he did not previously know about the documents Watson retrieved for his paper, and would surely include them as references in his next paper.

4.2 Related Work

Software that recommends Web pages and learns user preferences has been an intense focus of most recent research. Closely related work includes Metasearcher [2] a system which uses a collection of browser caches gathered from users working in collaboration on a common research task to form queries that are sent to search engines, the results of which are analyzed using LSI [13]; and Remembrance Agents [29, 30], systems that suggest similar documents as a user composes a new document by performing IR search against a local corpus of previously written text. Personal profiling agents (e.g., [7, 24]) are also related to our work on Watson. In addition, several recent commercial offerings have that is somewhat similar to ours. Autonomy's Kenjin system, released early in 2000, and a system called Nano, released later that year, both make recommendations for further information based on an analysis of the active documents and activities of the user.

5. I2I: Supporting Awareness of Potential Collaborators

I2I (first described in [4]), like Watson, is an instantiation of the architecture and approach described above. Like Watson, I2I tracks the documents users manipulate in and attempts to find resources that could be useful in the context of the work they are doing. The focus in I2I, however, is on providing users with access to people that could be helpful to them as they work.

Many efforts in building collaborative systems have focused on developing techniques to support awareness among individuals engaged in collaborative tasks (e.g., [18, 35]). While tools that support group awareness in the context of inherently collaborative work (e.g., meeting scheduling, collaborative design, etc.) are becoming more common, less attention has been given to supporting the kind of awareness necessary to give rise to collaboration in the first place—the kind of informal collaboration and communication that commonly occur in physically-located settings organized around common goals (as Kraut, et al. [22], Kristoffersen and Ljungberg [23], and others have observed). The goal is to build this kind of awareness of opportunities for collaboration into software systems in order to support interactions among people that are opportunistic and are often based on a establishing a shared context for the interaction.

The successes of collaborative systems like electronic mail have demonstrated computers connected to networks across significant distances can successfully mediate communication. Electronic "places" such as Web sites or newsgroups have evolved as landmarks for communities of interest that serve as the primary locus of collaboration and sharing for a community. Our view is that the potential value of such communities is frequently left untapped because of the friction associated with discovering them. The effort required to discover online communities is analogous to the effort required to

discover online documents. Our approach is to bring people together based on the work they are currently doing. Instead of requiring the user know what people, which community, or what “place” to go to, the places, the communities and the people are brought to the user, based on their relevance to her current work.

Our aim is for I2I to make opportunities for informal collaboration more obvious and more pervasive by reducing the amount of work necessary to become aware of them. The motivation is the possibility that by increasing awareness of common interests, users will leverage each other’s knowledge and experiences more frequently, which could allow them to be more productive. The problem is finding a balance between the benefits of having access to a large, diverse body of people and the level of effort necessary to find someone helpful. On the one hand, access to large numbers of people means someone relevant is probably out there. Current electronic communication systems give us that. Unfortunately it could take years to find the right person through traditional methods, a cost which often far outweighs the benefits users expect. I2I is designed to automate part of this process by noticing opportunities for collaboration based on the work people do in everyday applications. It provides a first cut at helping users discovering potential informal collaborators by giving users opportunities to become aware of others who are working on similar documents. Combined with standard communication tools, our goal is for a system like I2I to routinely transform traditionally solitary activities into collaborative ones by providing its users with frictionless access to potentially relevant others.

I2I automatically tracks the content of documents users manipulate using standard productivity applications. It clusters these documents based on their content, grouping related documents into a conceptual neighborhood, allowing users to:

1. Establish synchronous communication with others who are manipulating related documents.
2. Initiate conversations asynchronously through a facility we call calling cards.
3. Browse related information items automatically recommended by the system.
4. Join or start public chat rooms associated with the content area in which they are situated.

I2I attempts to manage the early stages of initiating informal collaboration by providing its users with opportunities to become aware of the activities of others that share common interests, as represented by the documents they interact with. I2I attempts to build communities of common interest on the fly, allowing users engaged in traditionally solitary activities to discover common goals and collaborate with each other, while reducing the overhead of orchestrating collaboration.

5.1 Related Work

Tools that allow users to collaborate around common electronic artifacts have been studied extensively, although much recent work has focused on collaboration around documents. Anchored Conversations [9], for example, allow collaborators to easily distribute shared documents and situate conversations within the context of specific places in a shared document. Ensuring collaborators share the same artifact makes collaborative activities that depend strongly on artifacts (such as collaborative writing) easier.

Systems like this are aimed at supporting collaboration among users who already know each other and have a prior goal to collaborate. Our work intends to provide opportunities for users who may not know each other to collaborate informally by making opportunities for collaboration visible, and by automating the early stages of establishing collaboration (e.g., knowing who to talk to, and how to talk with them). In this way, the work we describe here is similar to Jung and Lee’s work on ePlace, a system aimed at providing a rich environment that supports mutual awareness among visitors of ecommerce sites [21]. This work has resulted in particularly clever visual representations of Web sites on top of which information is overlaid, indicating the presence of site visitors. Our work differs from theirs in that it strives to extend beyond using the (somewhat arbitrary) structural organization of the Web and Web sites to establish awareness. Instead I2I’s notion of location is based on the content of the information objects users manipulate. This allows the system to notice similar work contexts that are not explicitly associated by hyperlinks (or even published on the Web).

Other systems have examined the role of distributed, public artifacts (e.g., Web pages and online virtual environments) as shared contexts, allowing users who are manipulating or viewing the same object from distributed locations to communicate with each other, usually using text-based chat (e.g., [11, 16, 27]). This vein of work is the closest to the work we describe here. The main difference is that the above systems require objects tagged by unique identifiers (in the case of the Web, the page’s URL), and also that users manipulate the same object at the same time in order to collaborate. These requirements limit the opportunities for collaboration the system can make available, due to the sheer size of the Web and typical patterns of access. Web accesses have been shown to follow a Zipf distribution, which means it is unlikely two users will be on the same page at the same time except at the most popular sites [3]. The most popular sites are typically portals to other content, and hence are not ideal as a shared context for collaboration. Likewise, it is also clear that similar content can be found on many different URLs. As such, systems that use document identity as their only basis for introducing people limit the opportunities for collaboration they can notice. In addition, approaches that rely on public artifacts like Web pages exclude

unpublished electronic documents (e.g., the article I'm working on right now) from consideration. Together these issues limit the utility of such systems.

I2I attempts to overcome these problems by analyzing the content of the objects being manipulated by users in order to automatically cluster similar content. In effect, I2I builds a separate conceptual space, organized by the content of documents being manipulated, and then situates users and other information items in this space. In addition, I2I provides facilities for asynchronous communication, allowing users to notice opportunities for collaboration across time.

I2I is also related to matchmaking systems (e.g., [14, 19]), which introduce users with common interests to each other with the goal of building online communities and fostering community awareness. Work on matchmaking systems has generally focused on introducing users based on long-term interests represented in a user profile. In the Yenta system [14], for example, users submit a collection of documents to their personal agent, which builds a profile from those documents and executes a kind of distributed hill-climbing algorithm to match profiles using techniques from information retrieval. When a sufficiently similar profile is found, the agent arranges to introduce the two users. The matching algorithms used in such systems are typically designed to work asynchronously.

In contrast, our work on I2I focuses on introducing users based on their immediate (and perhaps short-lived) interests that arise from the tasks they in which they are currently engaged. Instead of requiring the users define a profile for themselves using documents or keywords, I2I automatically builds a lexical representation of the user's current activity (as represented by the document the user is manipulating) and uses this representation to determine what the user can see, as they are working.

5.2 Design and Implementation

Like Watson, I2I integrates with applications through their APIs and the operating system's inter-process communication facilities under the IMA architecture (described above). Each application has its own application adapter, which is responsible for communicating user actions and document content to a broker, located on a central server. The broker is responsible for persistent information such as the user's profile (e.g., their name, password, etc.), as well as ephemeral information, such as how to contact their machine, and the representation of the document they are currently manipulating. I2I also contains the functionality of Watson. In addition to providing the user with access to related people, it also provides her with access to related documents.

5.2.1 Brokering Opportunities for Collaboration

Each application adapter is responsible for sending the broker a message when the document has changed in an attached application (e.g., the document is edited significantly, or the user opens or navigates to a new one). The message the adapter sends to the broker (located on a central server or server cluster) contains the text of the active document, its location (URL), and the user ID of the I2I user.

The broker computes a term vector representing the current document under the vector space model [31], commonly used in information retrieval systems. Vectors are weighted using the *tfidf* heuristic [33], which values words that are frequent in the current document, but rare across the entire collection. The vectors representing each document correspond to points in a high-dimensional space; the number of unique words in the active documents determines the dimensionality of the space; the weighting heuristic determines the position of a vector in the space. Each user is associated with a vector (or several, in case they have multiple documents open). Associated with each vector is the title of the document it represents, the URL (if the document is a Web page), a list of users manipulating that document, chat rooms started from that document, as well as a list of calling cards associated with the document.

The broker computes a pair-wise similarity matrix for documents that are currently in use, represented by the cosine of the angle formed by the two vectors that represent them. Essentially, the cosine measure represents a weighted function of the features (in this case, word stems) the vectors (documents) have in common. The similarity matrix the broker computes is used to determine what a user sees from the vantage point of a particular document (see Figure 6). Objects associated with documents D whose similarity is above a threshold θ with respect to d , the current document, are visible from d (currently θ is set to 0.65 based on the experiments presented in the following section, but we are exploring interfaces that allow users to manipulate this parameter interactively).

The chance of two people reading exactly the same document at the same time may be slim. By grouping conceptually similar documents together, I2I makes it more likely that people will see each other and start a conversation. It also allows unpublished documents (e.g., a paper in progress) to serve as the entry point into the system.

Secondary objects can also be associated with a document in the space I2I has built. The simplest of these objects is people: users who are viewing a particular document are associated with that document's point in the space. Currently, I2I also indexes chat rooms and calling cards (a facility for asynchronous communication) in the same way. Users who access a document, then, can see the items associated with it and other documents close to it in the conceptual space (see Figure 6).

5.2.2 Client Interface

I2I tracks a user's current task context (represented by the document they are manipulating) so it can provide potentially useful resources to users in the context of a specific editing or browsing session. It embeds an interface for displaying this information directly into applications, where it is supported (see Figure 2) to allow the user to easily correspond the information I2I displays with the document it is associated with. In other cases, information is displayed in an associated window that can be "hooked" on to the main window of the application, to maintain visual correspondence. This allows users to easily keep track of their activity in several conceptual spaces at the same time, as was discussed in Section 3.

Details of the embedded interface are shown in Figures 2 and 7. Information is grouped into tabs and includes (from left to right):

1. *System activity.* Users can see how many I2I users are online both in and outside of the conceptual space defined by their document. Other activity information includes how many people are chatting, and how many related pages other I2I users are reading.
2. *Who is online.* Users can see the login names of the people reading or writing related documents, and pointers to the documents they are viewing, if they are available on the Web (see Figure 2). Users can contact each other directly via instant messaging, double-blind email, or by using videoconferencing software, depending on the software and hardware available on their machine.
3. *Related documents.* I2I displays related pages from other Web sites people are currently browsing. In addition, I2I displays recommendations generated by the Watson system (described earlier). The Watson system recommends related documents by automatically querying online information repositories.
4. *Active chat information.* I2I displays a list of chat topics created by users within the conceptual space defined by the current document. Users can also chat in a default room associated with this region in the content space.
5. *Calling cards.* I2I displays a list of calling cards that other users have left in the past while viewing the current or related documents (see Figure 7). A calling card is a note indicating that a user would like to talk about a particular topic.

5.2.2.1 Calling Cards

Users can leave calling cards associated with the content area represented by their document in order to indicate they would like to discuss a particular topic with other users. Figure 7 shows calling cards associated with a page about Richard Nixon.

Leaving a calling card allows users to make their goal to chat about a particular topic visible to other users who also view pages in that topic area. If a user is eager to open a discussion channel with somebody else, but no one is available or has shown interest, the user can leave a message to invite people to talk at a later date. After leaving a calling card, the user can continue to work, or even destroy the original document the calling card was associated with. Calling cards are indexed by a content vector representing the document at which they were created. This means access to the document is not necessary for other users to see them when they are browsing or writing in related areas.

For example, one user could leave a calling card at the document in Figure 2, which discusses Nixon and his presidency. That document could then be taken off of the Web. At this point, other users would still be able to see the calling card when they accessed other documents about Nixon, for example, the page in Figure 7.

This kind of indexing also nicely accommodates documents that have frequently-updated content (like the front page of a news site), because even though the content of a page might change, the system associates the calling card with the original context in which it was created. This ensures that the calling cards retrieved are actually relevant in the context of the

document being viewed. This approach is similar to the independently developed intra-document linking technique reported and evaluated in [28].

When another user sees the calling card, she can find out whether the owner of the calling card is online or not. If the owner is online, she simply drops a line to the user to say that she is interested in discussing the topic. If the owner is not online, she can find out if the owner has a public email address and send an email to the owner (if the owner has specified others can contact her via email).

A calling card persists for a time period specified when it is created (currently the system imposes a limit of 30 days). When a calling card expires, the owner is notified via the global interface. The user can then choose to delete the card, or extend the time period in which it is available.

5.2.2.2 Managing Privacy

Users may be uncomfortable, at times, having a system track what they write or view. I2I allows people to manage the privacy of their work by being highly visible when it is on (see Figures 2 and 7), and by allowing users to shut it off at any time (using the close button in the interface). I2I also does not expose the details of offline (non-Web) content to any third party. In addition, it does not expose a user's email address or online identity directly. Instead email is sent through a mediating server that automatically makes message sender and recipient anonymous. This allows users to disclose their real email addresses at their own discretion.

Pertinent future work includes allowing users to adjust the extent to which their identity is revealed by developing online trust relationships with other I2I users. The idea is that users who have no prior affiliation can choose to reveal elements of their user profile (for lack of a better term) to each other, even though the full profile could be used to contribute to the similarity score the system computes. In general, we anticipate mitigating privacy concerns will be a significant issue moving forward. As such, we are working on facilities that allow users to maintain awareness and control over their trust relationships with both the users they know and users they haven't met.

6.2.3 Global User Interface

I2I has a global interface that allows the user to control whether or not she is available for conversation, as well as edit her profile and preferences. Calling cards are also managed using this interface (owners of calling cards can edit or delete them at any time).

6.3 Evaluations and Analysis

I2I is still under development, and although it is a suitable demonstration system, it is not ready for deployment. Thus, in lieu of a field study, in which we could get a broader sense of the system's effectiveness, we have instead evaluated the matching techniques I2I uses on real data collected from users. We collected about two days of browsing logs from 11 users and performed an offline analysis. The browser logs were collected via a plug-in to Internet Explorer that recorded the URL, the time of access, and the content of a document when it was loaded. The users were graduate students at Northwestern (either in the School of Education or in the Computer Science department), or friends of the graduate students who participated.

Some of the following evaluations are performed by sampling the original distribution of accesses to simulate the use of the system by varying numbers of users. Because the distribution we observed follows larger distributions in character, the simulated results we describe are likely to be predictive of the system's performance with larger numbers of users. The one caveat is that the distribution of the content we collected may not match the content distribution of the Web in general, because the subjects who produced this data were not chosen at random from the collection of all Web users. We are not prepared to argue that the interests of graduate students are representative of the interests of the more general population of potential I2I users. However, one deployment strategy we are considering for I2I is aimed at facilitating awareness of the activities in others strictly within organizations. In this case, we would expect this data would be fairly representative (in distribution and structure, if not in content).

Note also that the content and frequency distributions of the data we gathered about users accessing documents on the Web is likely very different from their patterns of document access in word processors. In the future, we intend to extend these studies to include data gathered from users accessing and modifying documents in word processors.

That said, the analysis we performed attempted to achieve two goals. The first was to understand the relationship between the number of people using the system and the number of people they would see, on average, for varying levels of strictness

in similarity. This analysis allowed us to gauge the number of users the system requires to start providing contacts, essentially providing us with an idea of the “critical mass” requirements the system has (we use critical mass here as in [17]). The second goal was to evaluate whether a system using the techniques we describe above would make appropriate associations from the perspective of a potential user.

6.4.1 Summary of the Data Gathered

During the period we collected data, there were 1612 pages with unique URLs collected. This is only a lower bound on the number of unique pages viewed, because the same URL can contain different content (due to a form submission, for example). These pages were accessed a total of 5039 times.

As previous work (e.g., [20]) suggests, Web access data follow a Zipf distribution [36]. That is, if the frequency a page with frequency rank i is f_i , where the frequency rank i is the index of the i^{th} element in the sequence of documents accessed by descending frequency, then the Zipf’s Law states $f_i \propto i^{-\beta}$, where β close to -1 . The data we gathered follow this distribution, with $\beta = -0.79$ ($r^2 = 0.96$). The linear and log-log plots of this data displayed in Figure 8 are typical.

6.4.2 Critical Mass Analysis

The most important consequence of the fact that document accesses follow a Zipf distribution is that a large number of documents are accessed relatively infrequently for any given period of time. For the data we collected, 1322 pages were accessed below the mean frequency of 3.12 (that’s over 80%). About 50% of the documents were only accessed once. The data we collected supports our hypothesis that systems that provide users with information on who is browsing the *same* page would suffer serious “critical mass” problems. That is, at a given time, very few people will be present on anything but the most popular sites, leading to situations in which the system displays unmanageable numbers of people, or no one at all. The goal of the techniques we use for grouping people in I2I is to strike a more workable balance.

It is also interesting to note that the most popular pages in our set contain little lexical content (one of the top documents was the front page of a search engine with a markedly sparse interface). This follows results on a larger data set, reported in [10], which suggest smaller documents are accessed more frequently. More lexical content tends to improve the quality of match because words disambiguate each other’s meaning (see [32] for an early technique that exploits this). Fortunately, the data suggests the pages with the least amount of lexical content happen to be ones at which multiple users are most likely to be accessing at the same time. This means that even though the similarity metrics we use rely on lexical content, those documents with the least amount of lexical content are those most frequently visited and hence will be more likely to have visitors at the same time.

The first critical mass evaluation we performed was aimed at determining the relationship between the number of users in the system and the possibility they would see someone new if they came online. Our hope was that this would give us a general sense of how the system would perform with particular numbers of users. We expected that as the number of users increases, their coverage of the space would also increase (that is, there would be more pages for which a new visitor would see at least one other user).

To accomplish this, we produced iteratively greater random samples drawn from the original distribution of page accesses we collected, in order to simulate varying numbers of users. Then we computed the percentage of accesses that would be visible from the vantage point of each document in this set, given one of several similarity thresholds (again, drawing the next access from the original distribution). We repeated this 10 times for each sample of “users” and took the average. In addition, this same analysis was repeated for 10 thresholds. The results of this analysis are displayed in Figure 9. As expected, the analysis shows that looser thresholds cover more of the document space, and that as the number of users approaches the number of documents the space is also covered more. Figure 9 also shows the log-log plot of the same data which can be fit using least-squares regression with average $r^2 = 0.98$. The slopes are close to 0.5 for each threshold, and the x intercepts are increasing as the threshold increases. That is, as the threshold increases, the number of users that must be logged into the system before at least two users will see each other (the x intercept) also increases.

We also evaluated the relationship between the number of users using the system and the average number of people they could see. To do this, we performed the same random sampling from the original access distribution for varying numbers of people. We then computed the average number of people visible to each of these users given a fixed similarity threshold. We repeated this 100 times for each sample of “users” and took the average. We performed this analysis on 10 thresholds, for numbers of simulated users. The results of this analysis are displayed in Figure 10. Figure 10 (left) shows the raw data

for various thresholds. Perhaps more instructive are the regression lines displayed in Figure 10 (right) (the average r^2 is 0.94 for these fits). There are several interesting things about Figure 10 (right):

1. It shows that various clustering thresholds cause the number of people seen to diverge at different rates—faster for lower thresholds and slowest for the tightest ones. This is displayed graphically in Figure 11, which plots the slope of the regression line vs. the threshold.
2. It shows that clustering documents improves the chance of a user seeing someone, even at the strictest thresholds.
3. Third, it shows the number of active users needed to start reliably seeing other people from a particular document at a particular threshold. For loose similarity thresholds, this number is low. For higher thresholds, the number of active users required increases. This is displayed graphically in Figure 11 (right), which plots the values at which the regression line for a threshold crosses the line $y = 1$. This information will be valuable in the future, as we start to work on building newer versions of the system that can automatically adjust thresholds.

6.4.3 Appropriateness Analysis

In order to evaluate the appropriateness of the associations the system made, we had a volunteer uninvolved with this project evaluate the associations by hand. For each of 9 thresholds, we picked 10 random documents (we will refer to these 90 documents as the *source* documents). For each of the source documents, we randomly selected 10 documents that were similar above the threshold (the *target* documents). The volunteer was instructed to compare each source and the target and count the number of inappropriate associations made by the system (he performed a total of 900 comparisons). The results of this experiment are displayed in Figure 12. The data shows that for thresholds greater than or equal to 0.4, the system forms appropriate associations between source and target documents a least 60% of the time. For thresholds greater than or equal to 0.7, the system forms appropriate associations over 90% of the time.

6.4.4 Discussion

Together, the above analyses provide us with a better understanding of the relationships between number of people, similarity threshold, and relevance. Given an understanding of these relationships, we can begin to design the system to address the strengths and weakness revealed by this analysis. The above experiments were immediately useful in determining we should set the similarity threshold at about 0.6 or 0.7 in order to balance the tradeoff between the desire to have the system allow users to see other people, and the desire for the associations made by the system to be of the highest relevance. Future work entails determining what level of accuracy users find useful; this will help us further tune the system, evaluate the clustering techniques in this context, and develop new user interface facilities aimed at allowing the user to form correct expectations about how the system will operate.

It is important to realize we make a number of assumptions in this design. The most major assumption is that the text of the user's current document corresponds to her current goals and interests at a useful level of abstraction. We recognize this is not always the case, for example, when a user clicks on a link by mistake. However, our working hypothesis is that these are special cases.

The techniques we use for clustering documents have been shown to be effective (for example, similar techniques produce improvements in information retrieval [32]), but the above and other studies show that unintuitive associations can occur. It is also sometimes the case that the user's current document does not provide a very good window onto her goals (e.g., a single document can have multiple purposes). However, it is important to realize that the system does not require users to collaborate. It provides users with opportunities for collaboration by automatically recommending potential collaborators. In the end the user determines whether or not she takes the recommendation. Users can make their own decisions about whether to collaborate with each other based on their current needs and by inspecting the documents others are viewing (if they are available online), or by considering background information about the user (should users make such information available through the system). Improved interfaces for introduction are needed so that users who don't initially know each other can quickly determine whether or not spending the time to do so will be useful in the context of the tasks they are performing.

It is also important to note that the system provides the user with a representation of its current view of her work in the form of recommendations. If the system's recommendations are on-point, the user can be relatively certain the users recommended or the index terms for her calling card are also fairly appropriate. If the system displays off-point recommendations, then calling cards indexed in that context may be displayed in inappropriate places, and the users collected may be driven by unintuitive associations. Our studies show this will happen about 20% of the time [5]. The system's opt-in nature helps ameliorate the usability issues this causes. Likewise, examining the quality of the recommendations the system

gives can also serve as a benchmark against which users can build accurate expectations. However, future iterations of I2I will most certainly expose more of the system's internal representations of confidence to the user so that she can build better expectations about how the system will perform.

6.5. Ongoing and Future Work

We have distributed I2I to several researchers in our department for limited use. For the most part, the feedback was positive. Users said they like the sense of being in a community and enjoy the kind of ready connectivity brought by I2I. We were encouraged by this initial test and look forward to a larger-scale deployment effort.

In general, the system's usability is a concern. A second round of design is currently underway in which we plan a number of improvements aimed directly at addressing these concerns. After this round of design we plan to show the prototypes we develop to users. We have several improvements under way. We would like to allow users to better understand where a calling card will appear before a user commits to leaving one. Likewise, facilities for setting up groups or cohorts would assuage some of the privacy concerns we found users had when they initially began working with the system. We are also investigating interfaces that allow the user to narrow or broaden the conceptual spaces in which they are situated. And we are working on better interfaces for introduction.

We are also working on improving the techniques we use to automatically introduce users. The evaluations uncovered several technical difficulties with the document similarity techniques we use that make them inadequate for handling some Web documents (e.g., URLs should be included as "terms" in a document so that pages with a single client-side image map can be coherently handled). In addition, we are investigating techniques for filtering potential collaborators by profiles built from their long-term history of interacting with documents (so that people with similar backgrounds are preferred). Coupled with this we are working on automated techniques that will allow the system to dynamically adjust the similarity threshold in order to hold constant the number of users displayed. This will allow the system to more gracefully handle both the dense regions in the conceptual space, as well as the sparse ones we noticed in the data we collected.

Perhaps most compelling, however, is trying to understand what mix of people will benefit each other the most in the context of particular tasks. For example, a student stuck on a problem is likely to find another student who has finished that problem more helpful than someone who is also stuck. More generally, one aspect of a good collaboration is that it brings together people whose knowledge, skills, perspectives, and interests compliment each other in ways that are mutually beneficial. Giving the system better models of groups and individuals could allow it to automatically build this kind of complimentary collection of people. We see this as a particularly compelling direction for the future of this work.

7. Conclusion

The growing ubiquity of the Internet is changing the way people access information and the dynamics of how they interact with each other online. However, we can only take advantage of the resources available to us in this networked world if we are aware of them. The tools we described in this article are aimed at facilitating an awareness of the resources available online to a user in the context of her current task. Our hope is that these kinds of awareness cures can help users by reducing the friction required to access resources instrumental to their task. We described two systems that support resource awareness on the Web:

1. Watson, an assistant that provides its users with frictionless access to documents relevant to their current task that are automatically gathered from online repositories.
2. I2I, a system that embeds communication facilities in the user's everyday applications so that users who share common work contexts can become aware of each other and communicate, even though they may have never met or discussed the interests they share.

As we work to deploy these systems we are excited by their potential to positively change the way people work by allowing them to more easily leverage the resources available to them.

References

1. Ackerman, M., and Starr, B., "Social activity indicators: interface components for CSCW systems," in *Proceedings of The 8th ACM symposium on User interface and software technology*, (Pittsburgh, PA, USA), ACM Press, 1995.
2. Badue, C., Vaz, W., and Albuquerque, E., "Using an Automatic Retrieval System in the Web to Assist Co-operative Learning," in *Proceedings of The 1998 World Conference of the WWW, Internet and Intranet*, (Orlando, FL, USA), AACE Press, 1998.

3. Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S., "On the Implications of Zipf's Law for Web Caching," in *Proceedings of IEEE INFOCOM '99*, (New York, NY, USA)1999.
4. Budzik, J., Fu, X., and Hammond, K., "Facilitating Opportunistic Communication by Monitoring User Activity in Everyday Applications," in *Proceedings of The CSCW 2000 Workshop on Awareness and the WWW*, (Pittsburgh, PA, USA), ACM Press, 2000.
5. Budzik, J., and Hammond, K. J., "User Interactions with Everyday Applications as Context for Just-in-time Information Access," in *Proceedings of The 2000 International Conference on Intelligent User Interfaces*, (New Orleans, LA, USA), ACM Press, 2000.
6. Budzik, J., Hammond, K., and Birnbaum, L., "Information Access in Context," *Knowledge Based Systems*, 14(1-2), 37-53, 2001.
7. Chen, L., and Sycara, K., "WebMate: A Personal Agent for Browsing and Searching," in *Proceedings of The Second International Conference on Autonomous Agents*, ACM Press, 1998.
8. Chen, L. L.-J., and Gaines, B. R., "Methodological Issues in Studying and Supporting Awareness on the World Wide Web," in *Proceedings of WebNet '96*, (San Francisco, CA, USA), AACE, 1996.
9. Churchill, E. F., Trevor, J., Bly, S., Nelson, L., and Cubranic, D., "Anchored Conversations: Chatting in the Context of a Document," in *Proceedings of CHI 2000*, (The Hague, The Netherlands), ACM Press, 2000.
10. Cunha, C., Bestavros, A., and Crovella, M., Characteristics of WWW Client-based Traces, Boston University Computer Science Department Technical Report TR-95-010, June, 1995 1995.
11. Donath, J. S., and Robertson, N., "The Sociable Web," in *Proceedings of the Second International WWW Conference*, (Chicago, IL, USA), Elsevier, 1994.
12. Dourish, P., Edwards, K., LaMarca, A., Salisbury, M., "Using Properties for Uniform Interaction in the Presto Document System," in *Proceedings of UIST 99*, (Asheville, NC, USA), ACM Press, 1999.
13. Dumais, S. T., Furnas, G. W., Landauer, T. K. , Derrwester S. , and Harshman, R., "Using latent semantic analysis to improve access to textual information," in *Proceedings of CHI 88, The International Conference on Human Factors in Computing Systems*, ACM Press, 1998.
14. Foner, L., "Yenta: A Multi-Agent, Referral Based Matchmaking System," in *Proceedings of Agents 97*, (Marina del Rey, CA, USA) 1997.
15. Gamma, E., Helm, R., Johnson, R., Vlissides, J., and Booch, G., *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1994.
16. Goody, Goody, Available at <http://www.getgoody.com/>.
17. Grudin, J., "Groupware and Social Dynamics: Eight Challenges for Developers," *Communications of the ACM*, 37(1), 92-105, 1994.
18. Gutwin, C., and Greenberg, S., "The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware," *ACM Transactions on Computer-Human Interaction*, 6(3), 243-281, 2000.
19. Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S., and Yoshida, S., "Socialware: Multiagent Systems for Supporting Network Communities," *Communications of the ACM*, 42(3), 1999.
20. Huberman, B., Pirolli, P., Pitkow, J., and Lukose, R., "Strong Regularities in World Wide Web Surfing," *Science*, 280(5360), 95-97, 1998.
21. Jung, Y., and Lee, A., "Design of a Social Interaction Environment for Electronic Marketplaces," in *Proceedings of DIS 2000 - Designing Interactive Systems: Processes, Practices, Methods, Techniques*, ACM Press, 2000.
22. Kraut, R., Egido, C., "Patterns of Contact and Communication in Scientific Research Collaboration," in *Proceedings of CSCW 88*, (Portland, OR, USA), ACM Press, 1988.
23. Kristoffersen, S., Ljungberg, F., "An Empirical Study of How People Establish Interaction: Implications for CSCW Session Management Models," in *Proceedings of CHI 99*, (Pittsburgh, PA USA), ACM Press, 1999.
24. Lieberman, H., "Letizia: An Agent That Assists Web Browsing," in *Proceedings of The International Joint Conference on Artificial Intelligence*, (Montreal, Quebec, Canada), AAAI Press, 1995.
25. Lieberman, H., "Integrating User Interface Agents with Conventional Applications," in *Proceedings of The 1998 International Conference on Intelligent User Interfaces (IUI 98)*, (San Francisco, CA, USA), ACM Press, 1998.
26. Olsen, D., Hudson, S., Verratti, T., Heiner, J., and Phelps, M., "Implementing Interface Attachments Based on Surface Representations," in *Proceedings of CHI 99*, (Pittsburgh, PA), ACM Press, 1999.

27. Palfreyman, K., and Rodden, T., "A Protocol for User Awareness on the World Wide Web," in *Proceedings of CSCW 96*, (Cambridge, MA USA), ACM Press, 1996.
28. Phelps, T., and Wilensky, R., "Robust Intra-document Locations," in *Proceedings of Ninth International World Wide Web Conference*, (Amsterdam, The Neatherlands), Foretec Seminars, 2000.
29. Rhodes, B., and Starner, T., "A continuously running automated information retrieval system," in *Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology*, (London, UK), 1996.
30. Rhodes, B., "Margin Notes: Building a Contextually Aware Associative Memory," in *Proceedings of The 2000 International Conference on Intelligent User Interfaces*, (New Orleans, Louisiana, USA), ACM Press, 2000.
31. Salton, G., Wong, A., and Yang, C. S., "A vector space model for automatic indexing," *Communications of the ACM*, 18(11), 613-620, 1971.
32. Salton, G., and Buckley, C., "Improving Retrieval Performance by Relevance Feedback," in: Spark Jones, K., and Willett, P., ed., *Readings in Information Retrieval*. Can Francisco, CA: Morgan Kauffman, 1997.
33. Salton, G., and Buckley, C., "Term-Weighting Approaches in Automatic Text Retrieval," in: Spark Jones, K., and Willett, P., ed., *Readings in Information Retrieval*. San Francisco, CA: Morgan Kaufmann, 1997.
34. Selberg, E., and Etzioni, O., "The MetaCrawler Architecture for Resource Aggregation on the Web," *IEEE Expert*, 12(1), 8-14, 1997.
35. Whittaker, S., Swanson, J., Kucan, J., and Sidner, C., "Telenotes: Managing Lightweight Interactions in the Desktop," *ACM Transactions on Computer-Human Interaction*, 4(2), 137-168, 1997.
36. Zipf, G., *Human Behavior and the Principle of Least-Effort*. Cambridge, MA, USA: Addison-Wesley, 1949.

FIGURE LEGENDS

Figure 1: Watson has retrieved related items for a user reading a Web page about China and Taiwan.

Figure 2: People visible from a page about Nixon include users who are viewing different documents on the same topic.

Figure 3: The Information Management Assistant architecture for coupling agents, existing applications, and online information resources.

Figure 4: An interface attachment for the Watson system. The top three dots (A) represent the system's run status. The bottom five (B) represent the relevance of the information the system has gathered. When the user clicks on the minimized interface (left) she can access the full interface (right).

Figure 5: Watson (unattached) is displaying related images while a user is working on a paper about Citizen Kane.

Figure 6: I2I builds a conceptual space based on the content of documents, regardless of their location on the Web.

Figure 7: Calling cards visible from a page about Nixon.

Figure 8: Distribution of access frequencies by frequency rank. The graph on the left shows the data plotted on a linear scale, the data on the right shows a log-log plot.

Figure 9: Left: number of simulated people (x axis) vs. likelihood an additional person will become visible if one is added (y axis) at 9 thresholds (series). Right: the same graph plotted on a log-log scale.

Figure 10: Number of simulated people vs. average number of people they would see from a particular page. Left: actual data; Right: best fit lines (least squares regression).

Figure 11: Left: Slope of the regression lines in Figure 12 vs. threshold. Right: number of people that must be using the system to see one other person on average, vs. threshold (interception of the regression line with $y = 1$).

Figure 12: Threshold vs. percentage of inappropriate associations made by the system. As expected, as the threshold increases, the number of erroneous associations decreases.

FOOTNOTES

1. In many cases, especially for large software applications, having to resort to modifying an application's source code is not ideal or desirable anyway.
2. In Microsoft Windows, there are several techniques for doing this. One is called "installing a hook procedure," in which a user-defined event processing function is installed in the system's message processing chain and is able to intercept events for a window. Another is called "instance subclassing", in which a user-defined event processing function replaces a window's event processing function (with the option of passing messages along to the replaced function, much like subclassing in object-oriented programming languages). We have built versions of the interface attachment that use both techniques, but have found the latter one results in a better user experience because the operating system performs smoother rendering.

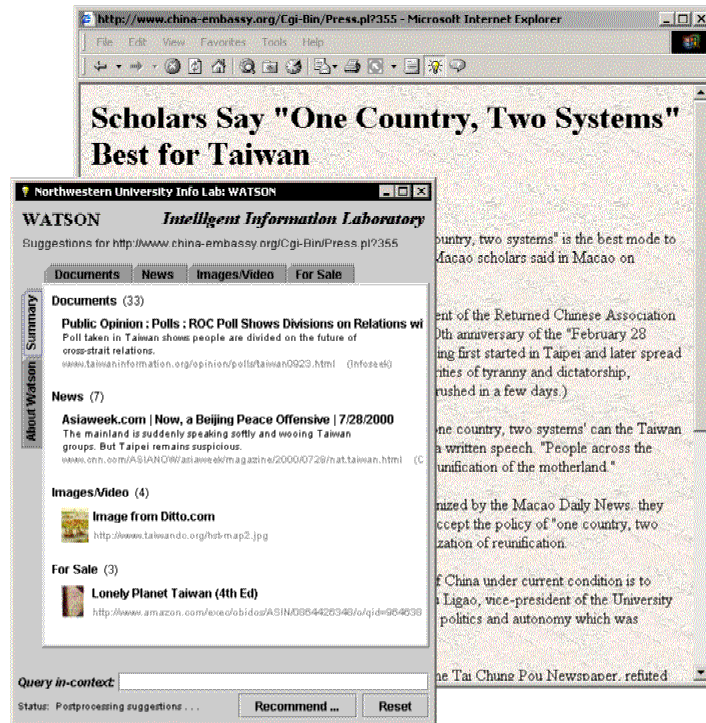


Figure 1
This figure should fit in a single column.

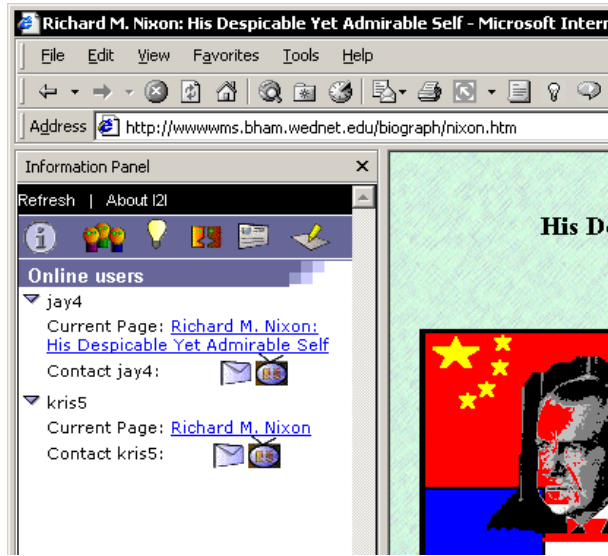


Figure 2
This figure should fit in a single column.

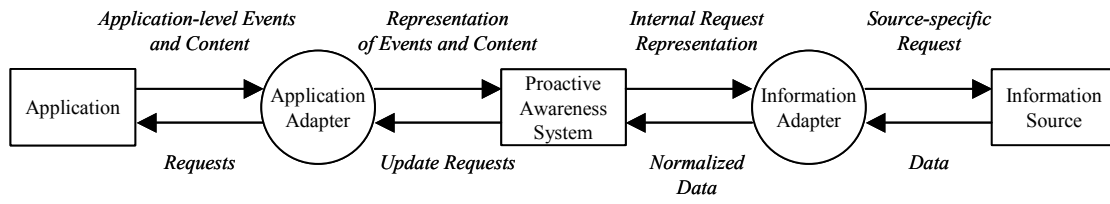


Figure 3

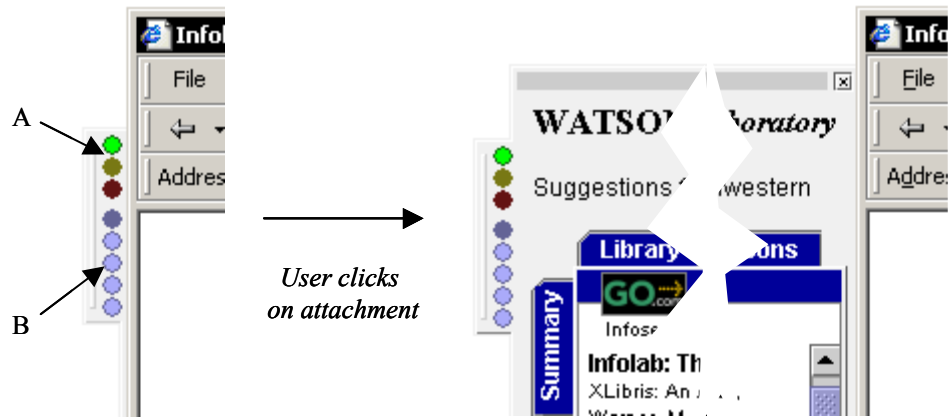


Figure 4

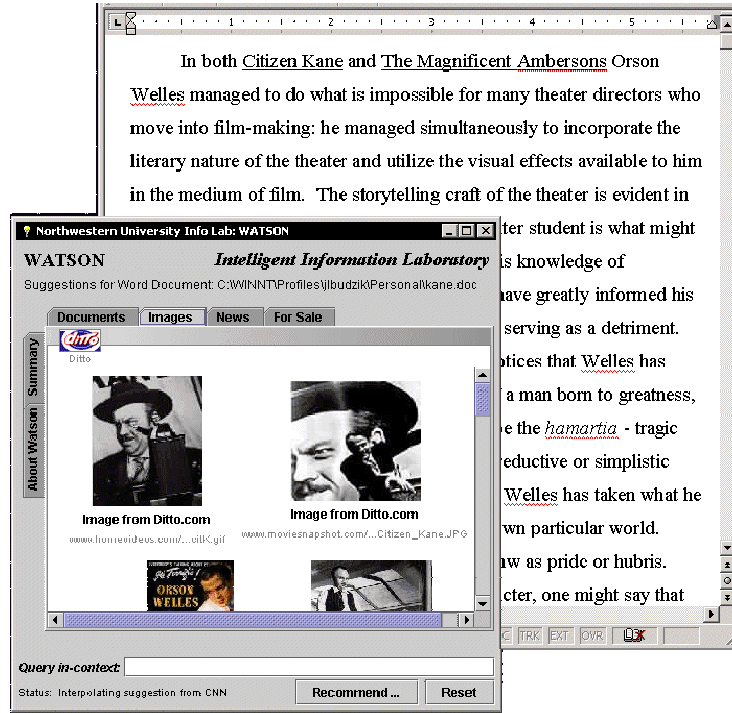


Figure 5

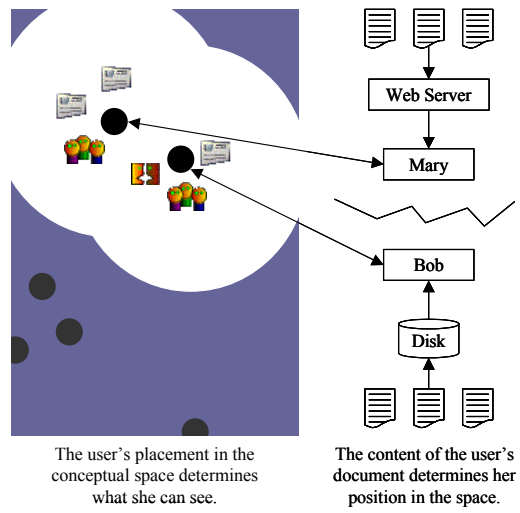


Figure 6
This figure should fit in a single column

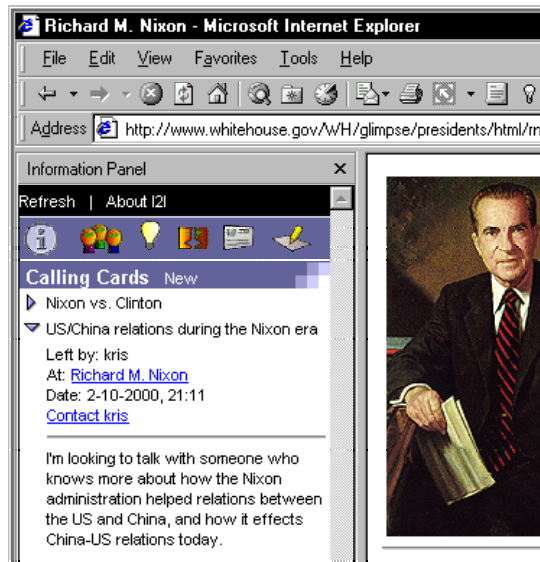


Figure 7
This figure should fit in a single column.

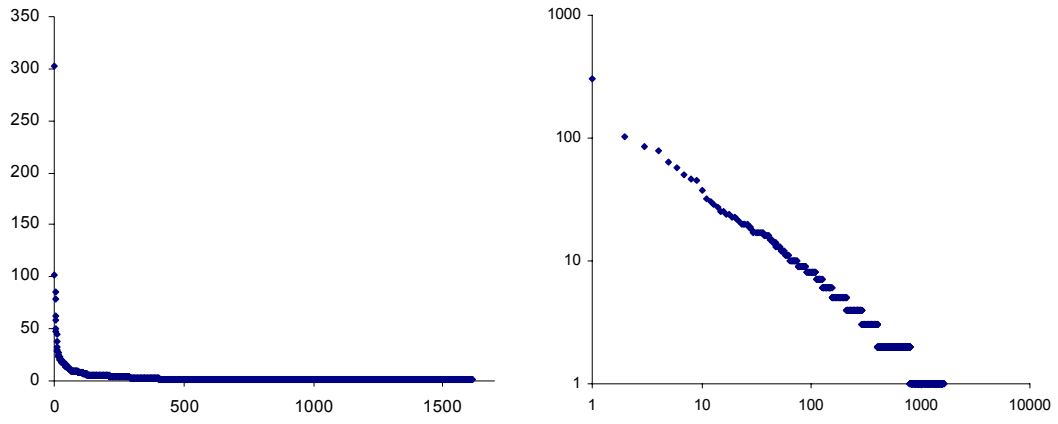


Figure 8

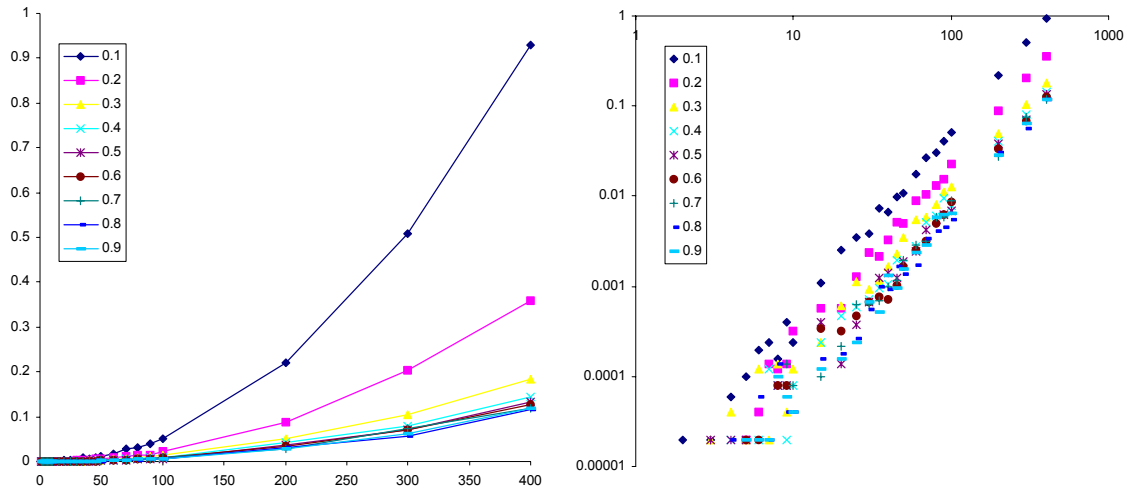


Figure 9

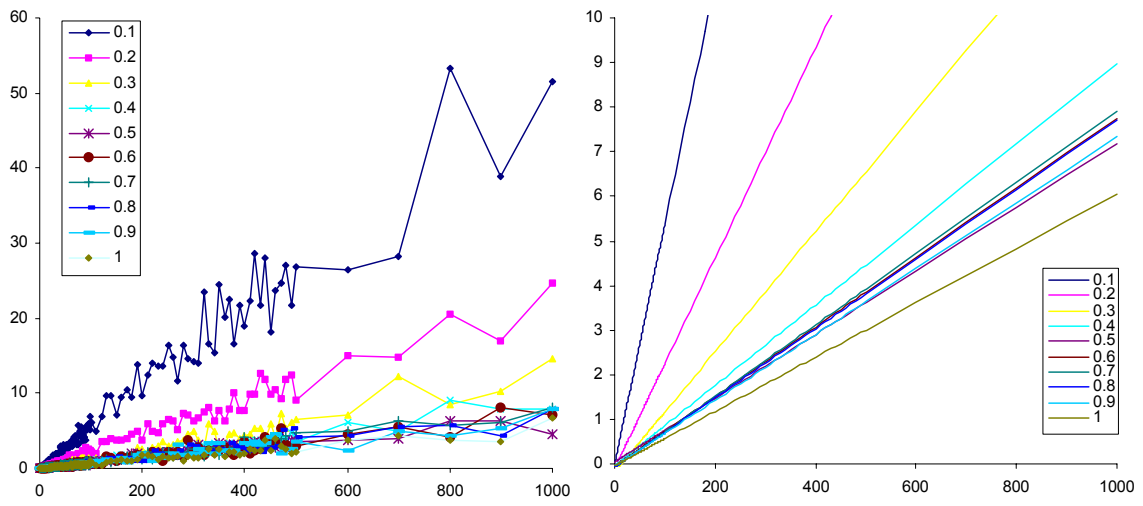


Figure 10

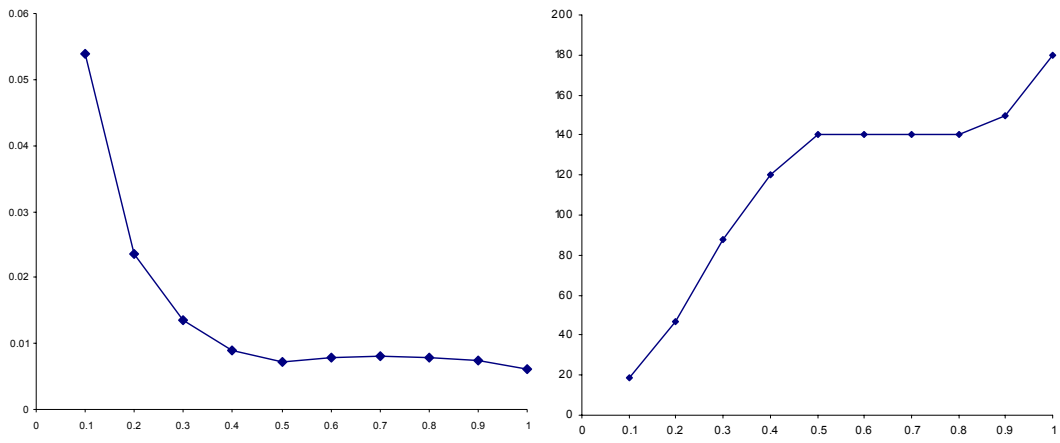


Figure 11

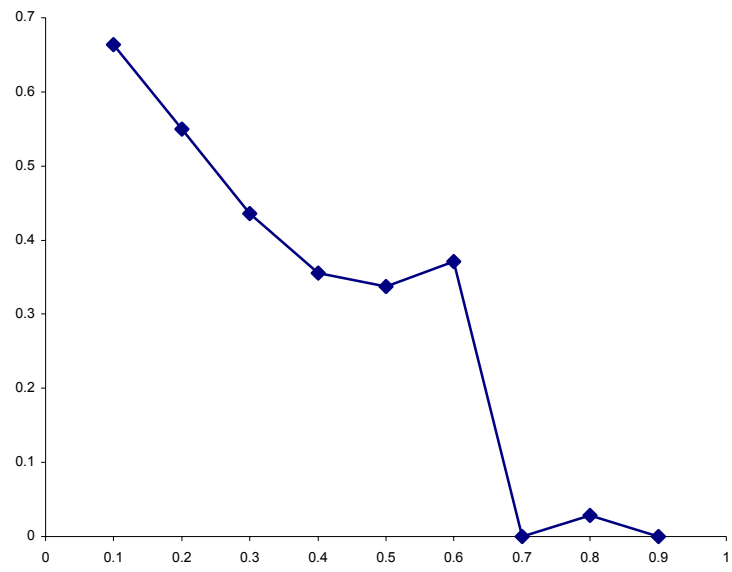


Figure 12