2009

Intelligent User Interfaces with Adaptive Knowledge Assistants

Adam Eck
University of Nebraska-Lincoln, aeck@cse.unl.edu

Leen-Kiat Soh
University of Nebraska-Lincoln, lsoh2@unl.edu

Intelligent User Interfaces with Adaptive Knowledge Assistants

Adam Eck and Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska-Lincoln

1. Introduction

One promising application of intelligent user interfaces (IUIs) is the support of daily tasks for information workers. In such a scenario, users perform various general and domain-specific information generation, processing, and retrieval tasks (e.g., processing data, creating reports, and searching for previous results). Intelligent user interfaces can improve the daily lives of such workers through activities such as task management, including managing the user's todo list and schedule (Conley and Carpenter, 2007; Myers et al., 2007), as well as summarizing meeting notes (Basu et al., 2008) or recommending tasks for the user to perform (Cosley et al., 2007) or users to collaborate with (Guy et al., 2009).

In this paper, we focus on a specific class of information workers: those specializing in using information systems to generate *knowledge*, which we term knowledge workers. While knowledge workers might perform some of the same tasks as other information users, e.g. organizing email communications from other users and meeting schedules through personal information management software, knowledge workers also perform tasks whose primary intention is to create new knowledge from information and data contained in the information system, used to both inform future activities and advance the current state-of-the-art in their domain. For example, a biologist working with a set of bioinformatics tools, e.g. BLAST (Altschul, 1990), to investigate genetic data stored in a database is a knowledge worker, as is a mechanical engineer studying the effect of tension forces on a skyscraper design.

We consider the environment inhabited by knowledge workers, pictured in Figure 1, to consist of a set of (possibly collaborating) users and a knowledge system composed of databases which store raw data, processed information, and generated knowledge, as well as tools, i.e. applications, for creating knowledge from the contents of those databases. The applications within the knowledge system can interact with the external world to gather information, e.g. overseeing an automated meteorology experiment to extract data from sensors distributed throughout a storm, and raw data from the external world can periodically filter into the system's databases. Users can perform tasks using applications to produce new knowledge, as well as query databases for contained information and knowledge. This interaction with the system takes place through the use of an intelligent user interface which performs support tasks such as those outlined in the previous paragraph. While a traditional user interface might be sufficient for providing the bare essential functionality required from knowledge workers to accomplish their tasks within the environment, the added intelligence within the interface allows the system to provide personal support tailored to the individual needs of each user, such as the aforementioned benefits of task management and recommendation, summarization, and finding appropriate collaborative partners. These benefits ease the burden placed on users to perform tasks and aim to boost user productivity and efficiency.

Given such an environment, the key question we are interested in is "How should we provide support for knowledge workers through the intelligent user interface for their daily tasks?" Specifically, we focus on a framework utilizing a variety of intelligent user interface technologies, coupled with personal support agents, to support knowledge workers in a scientific investigation and discovery application. We have developed a framework for Adaptive Knowledge Assistants (AKA) as personal assistant agents

Department of Computer Science & Engineering, University of Nebraska-Lincoln, Technical Report, TR-UNL-CSE-2009-0011

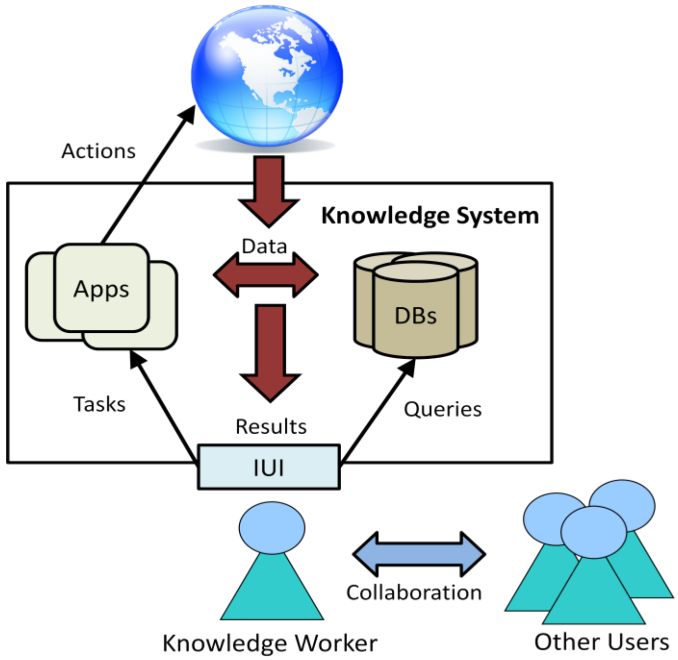

Figure 1: Knowledge System Environment

for such knowledge workers, which we propose in this paper. These agents utilize various components, including user modeling, task management, information management, group support, and user training, to improve user productivity and the end-user experience. We relate this framework to an actual knowledge system and describe how the framework follows the state-of-the-art in IUI research.

The rest of this paper is organized as follows. First, we provide a survey of recent trends in both personal assistant agent and intelligent user interface research, focusing on the areas most relevant to our work. Next, we present our Adaptive Knowledge Assistants framework for personal assistant agents supporting knowledge workers, along with the challenges it is designed to address and a comparison with surveyed IUI research topics. Finally, we describe the Semantic Cyberinfrastructure of Investigation and Discovery along with the related Biofinity project, a knowledge system designed to support scientific discovery and investigation. Here, we focus on the intelligent user interface needs of the application and present how we can apply the AKA framework to support the activities required of the intelligent user interface within SCID/Biofinity. We conclude with a brief summary and an outlook on our future work.

2. Background

In order to provide intelligent support to knowledge workers through an intelligent user interface, we utilize personal assistant agents augmented with a variety of general intelligent user interface technologies. In this section, we first introduce the concept of personal assistant agents, followed by an introduction to the supporting intelligent user interface technologies. For both, we provide an overview of the general ideas studied within each area, along with some recent research directions, especially those focused on supporting knowledge workers. These reviews are based primarily on the most recent (2007-2009) proceedings of the Intelligent User Interfaces (IUI) and Human Factors in Computing Systems (CHI) conferences, with some references drawn from other related conferences and journals. Please note that these reviews are not in-depth analyses of the content of the individual areas, but more of a broad introduction of concepts relevant to our design of Adaptive Knowledge Assistants and their application to supporting users of the SCID infrastructure and the related Biofinity application.

2.1. Personal Assistants

One of the primary purposes of employing an intelligent user interface within an application is to provide adaptive support to user needs. Some interfaces and applications go beyond this central idea to also provide a personal assistant to each user. This assistant is an intelligent software agent capable of learning about its supported user by observing her behavior within the software environment, making decisions to provide intelligent support, and acting on those decisions to change the software environment and assist users, e.g. performing tasks or monitoring events (Maes, 1994). For example, in a collaborative groupwork application, a personal assistant agent might profile its user, track the tasks performed by the user and notify her when new information is available or events are scheduled to occur, as well as find collaborative partners matching the user's tasks and complimenting the user's abilities. The goal of using such personal assistant agents is to reduce the burden on human users in their task completion and information management and improve user productivity.

Various personal assistant systems have been developed over the years. Maes (1994) created a personal assistant agent to manage a user's email inbox. Horvitz *et al*. (1998) studied the use of Bayesian models for determining user needs in the Lumiere project, the predecessor of the Microsoft Office assistant. Chalupsky *et al.* (2001) created the Electric Elves project using personal assistant agents to manage users' schedules, establish meetings, and track information such as users' flights. Through this project, the notion of adjustable autonomy was studied, representing the need of a personal assistant agent to adjust its autonomy during decision making based on its relationship to the human user to avoid overstepping the agent's responsibilities (Chalupsky et al., 2001; Tambe et al., 2006). More recently, researchers at SRI have studied personal assistant agents under the Cognitive Assistant that Learns and Organizes (CALO) project. In CALO, agents act as personal execution assistants (PExA) used to manage a user's schedule in a calendar (Myers et al., 2007) and manage tasks on a user's to-do list (Conley and Carpenter, 2007).

Recent research on personal assistants has focused on learning workflow procedures executed by users, both for modeling user tasks and potentially automating execution of tasks by agents in the future (Shen et al., 2009; Yorke-Smith et al., 2009). This fits within CALO's goal of acting as a proactive personal execution assistant for user tasks (Yorke-Smith et al., 2009) while maintaining Electric Elves's adjustable autonomy between agent and human (Chalupsky et al., 2001), i.e. only automating when allowed by a user. Other recent research related to personal assistants for knowledge workers has been conducted along the lines of assisting human users to search through distributed databases, including work by Tuchinda *et al*. (2007) studying how to create new interfaces for executing user queries across multiple data sources while abstracting access to the sources. Using this approach, a personal assistant can execute queries for users without requiring users to know how to interact with the various data sources searched with the query. Thus, personal assistants can not only perform tasks for users, but they can provide a simplified interface by performing low-level work for users inputted at higher abstraction levels more related to the way users think about their tasks.

2.2. Intelligent User Interface Technologies

Within the intelligent user interface literature, several general areas of recent research stand out, especially with respect to supporting researchers during scientific investigation and discovery through personal assistant agents. Here, we describe six: 1) recommendations, 2) intelligent notification/interruption, 3) preference elicitation, 4) learning from human users, 5) summarization, and 6) collaboration.

2.2.1. Recommendations

One of the most established areas related to intelligent user interfaces is the study of making recommendations, usually studied in the context of recommender systems. Here, the goal of the system is to provide recommendations of items, e.g. products, resources, people, queries, for users to examine, purchase, etc (Adomavicius and Tuzhilin, 2005). The key problem in the study of recommender systems is how to use a model of user behavior, e.g. a user's search history, liked and disliked items, and similarity to other users, to predict which items a user might prefer. This problem leads to several important challenges, including how to elicit user preferences (c.f., Section 2.4), how to represent user behavior, and how to compare similar users. Presently, there are three main approaches to solving the problem of predicting items to recommend: 1) content-based algorithms, 2) collaborative filtering algorithms, and 3) a hybrid approach combining the other two. First, content-based approaches choose items to recommend based on their similarity to ones the user has already preferred. For example, in an online bookstore application, if the user has shown a propensity to purchase books by a certain author, a recommender system might recommend other books by the same author, or by authors of the same genre. This approach relies on models (i.e., profiles) of items in order to make similarity comparisons, as well as a history of items preferred by users. The advantages of this approach include a lower reliance on user models (represented as a history of ratings or selections), simplifying preference elicitation and user modeling. However, the primary disadvantages of content-based algorithms include overspecialization, where the algorithm will only recommend items similar to those already seen by users, creating a lack of variety in recommendations, and limited content analysis, where the accuracy of recommendations requires complete, in-depth models of items for proper comparison (Adomavicius and Tuzhilin, 2005).

Second, instead of relying on the similarity between items for recommendations, collaborative filtering algorithms instead use the similarity between users for creating recommendations. In general, such an approach first finds users similar to the one being supported, then selects items preferred by the similar users to recommend. In our bookstore example, a collaborative filtering algorithm could find users with similar purchase or browsing histories to the local user, then recommend books purchased or highly rated by the other users. This approach overcomes challenges such as overspecialization since the system no longer just recommends items similar to those already preferred by users, instead pulling from a potentially wide range of preferences from other users. It also avoids the need for complex item models since item similarity is no longer computed. However, the drawbacks of collaborative filtering approaches include requiring more complex user models for accurate similarity comparisons, as well as sparsity, where many items in the system are likely to have been rated or preferred by at most only a few users, limiting their likelihood of being recommended in the future (Adomavicius and Tuzhilin, 2005).

Finally, hybrid approaches combine the positive aspects of content-based and collaborative filtering approaches to improve the recommendations offered by the system. These approaches range from simple combinations like simultaneously performing both content-based and collaborative filtering to select items, then alternating recommendations from each engine, to "demographic filtering", where not just user histories are compared in a collaborative filtering approach, but also background information on users is compared like items are in a content-based approach, to using similarity metrics that account for both user similarities when finding items to recommend in conjunction with similarities between items to create joint recommendations (Adomavicius and Tuzhilin, 2005).

Recent fundamental research in recommender systems has focused on several topics, including improving the accuracy of collaborative filtering with more complex models of user history and behavior (Bonnin et al., 2009) and choosing better metrics than just accuracy for both evaluating recommender systems and selecting items to recommend, e.g. user satisfaction (Hijikata, 2009). Other research has studied the use of information from social networking and Web 2.0 standards to improve recommendations, such as social tags, i.e., short textual descriptions or keywords attached to items by communities of users (Vig et al., 2009). Application-oriented research centered around knowledge workers (such as SCID and Biofinity) has investigated the use of recommender systems for choosing resources, e.g. documents, URL links, for knowledge workers based on current tasks and activities (Shen et al., 2008), selecting community Wiki articles for users to edit based on previous edits, links between articles, article titles, and similar author interests (Cosley et al., 2007), and finally suggesting people to collaborate with in social networks (Guy et al., 2009).

2.2.2. Intelligent Notification/Interruption

In order to provide intelligent support for users, interfaces need to periodically ask for user input and/or provide notifications of events, e.g. preference elicitation, notification that a new message arrived or an automated task was completed. However, each of these interrupts the user's current task, potentially disrupting the user's cognitive state, raising stress, and lowering productivity (Mark et al., 2008). To minimize distractions and these negative consequences, some intelligent user interface researchers have studied the problem of intelligent notification/interruption which aims to predict optimal times for interruptions and provide notifications in the most productive manner. For example, when supporting a knowledge worker with a report writing assignment, the user might be better interrupted when they complete a section of the report, rather than midsentence or paragraph. Identifying and reasoning about these moments of interruption is one approach to intelligent notification/interruption. Iqbal and Bailey (2008) take this approach and created a system capable of detecting breakpoints in user tasks based on switches between tasks at various levels of abstraction, e.g. in our report example, moving the cursor between paragraphs versus switching between application windows. Depending on which levels of abstraction breakpoints occur at, these breakpoints can represent ideal times to interrupt users. In an empirical study, this approach was well received by user participants, who especially liked relevant interruptions at the finer levels of abstraction, e.g. paragraph switching, and less relevant interruptions at higher levels of abstraction, e.g. application switching. Additionally, Hui *et al.* (2009) take a different approach and instead use a model of user mental states (based on probability distributions) to estimate user disruption from interface changes, which could possibly be extended to predicting notification disruption.

From the notification model perspective, Gluck *et al.* (2007) studied the effect of different types of notifications when interrupting users. Based on the work of Obermayer and Nugent (2000), they hypothesized that notifications whose observed importance is proportional to the utility of the interruption, e.g. urgent messages from colleges have higher expected utility than an email from an unknown party, would result in fewer negative consequences from the interruption. This hypothesis was confirmed in an empirical study of different notification techniques, including small flags placed on icons, icon pulsing, and icons that follow the user's mouse cursor for maximal attention.

Other empirical studies supporting research on intelligent notification/interruption have investigated both human beliefs about interruption and the effects of using strategies to manage notifications to users. Specifically, Avrahami *et al.* (2008) studied the relationship between a person's belief about their interruptability and an observer's similar belief based on what they observe about the other's behavior. They found that while the two are often correlated, humans often rate themselves as more interruptable (i.e., less consequence of interruption) than observers. This study also identified key misconceptions in interruption estimation which could be useful for designing intelligent notification systems. Mark *et al.* (2008), on the other hand, studied the general effects of an intelligent notification/interruption system on user behavior. Here, they confirmed that interruptions do raise stress levels of knowledge workers, but unexpectedly, they also observed that interruptions caused users to complete jobs faster, possible from a heightened sense of urgency due to higher stress levels and less time available for completing their tasks.

2.2.3. Preference Elicitation

When making decisions to support users, an intelligent interface must be able to reason about the preferences of users in order to provide optimal support from the user's perspective. Extracting from users, representing, and reasoning about these preferences is the problem of preference elicitation (Peintner et al., 2008). Preferences can be elicited from users in multiple ways (Chen and Pu, 2004; Pu and Chen, 2008), including asking users to choose a preference between two alternatives, "tweaking" a best query match to find related alternatives (Burke et al.,1997), rating items in recommender systems (Hu and Pu, 2009), and predicting preferences based on a user model, such as a personality quiz (Hu and Pu, 2009). Challenges in preference elicitation include revising preferences to match changing user interests (Pu and Chen, 2008), creating a strategy to determine preferences, e.g. modeled as a sequential decision problem (Boutilier, 2002), and choosing a representational model for preferences, e.g. as factored utility functions (Braziunas and Boutilier, 2008). Reasoning about preferences is application-specific and tied to the use of preferences in intelligent user interfaces, e.g. recommending items in a recommender system.

To determine user preferences over preference elicitation strategies, Hu and Pu (2009) conducted an empirical study comparing two strategies for eliciting preferences in a recommender system setting: explicit item ratings and personality quizzes. Here, ratings describe how much a user likes an item on a numeric scale, while the results of the personality quiz are used to predict preferences over a set of items. The tradeoff between these approaches is direct measurement versus user effort: while ratings directly measure a user's preference of an item, every single item must be rating to elicit the proper preferences from users. Personality quizzes, on the other hand, indirectly predict preferences based on some relationship model between personalities and items, but can generalize across unseen items. Hu and Pu observed that due to the less effort required for preference input through personality quizzes, user cognitive effort was lower for this technique, and user loyalty, i.e. the likelihood that the user would use the system again or recommend it to others, was higher for personality quizzes. Interestingly enough, even though personality quizzes indirectly measure preferences over items, the recommendations provided by the two types of elicitation were not significantly different in terms of accuracy (although personality quizzes again came out slightly ahead). This work demonstrates the advantages of considering the impact on users of preference elicitation strategies and is related to the aforementioned research on intelligent notification/interruption in that a strategy for determining user preferences should minimize disruption from other tasks and avoid burdening user's cognitive loads.

2.2.4. Learning from Users

Another area of research within the intelligent user interface literature (and studied elsewhere in the machine learning and intelligent agents communities) relevant to supporting knowledge workers is learning from humans. Specifically, the goal of this research is to provide algorithms, methodologies, and approaches for improving performance of computational systems based on user feedback, as well as learning how to perform tasks or creating beliefs from human instruction or observation. This broad range of topics can be divided into three main categories: 1) programming by demonstration (Cypher, 1993), where a user programs a computer not through written programming languages but by demonstrating tasks which are observed, generalized, and learned by the system; 2) bootstrapped learning (Oblinger, 2006), where computers learn concepts and procedures directly from natural human instruction based on prior knowledge; and 3) interactive machine learning (Fails and Olson, 2003), where computers train machine learning classifiers based on interactions with users. Please note that this area of research differs from learning *about* human users, which falls under the domain of user modeling and preference elicitation. Rather than concentrating on the users, learning *from* human users focuses on learning about some task, concept, etc. based on information from users.

Recent research in the area of learning from humans include simplifying web pages for mobile devices based on extracting important interface elements from user interactions with full versions of the same web page (Nichols and Lau, 2008), the MABLE framework as workbench for testing the ability of algorithms to learn from natural human instruction in bootstrapped learning (Mailler et al., 2009), and debugging learned programs from user observation (Chen and Weld, 2008; Kulesza et al., 2009). More specific to supporting knowledge workers such as those in SCID and Biofinity include learning dataflows for completing tasks from observing human users in order to provide task-aware support and automated task execution (Gervasio and Murdock, 2009), learning how to select resources based on complex requirements using feedback from users rather than complex queries (Ritter and Basu, 2009), learning how to dialog with users based on human-human dialog logs (Orkin and Roy, 2009), and programming by demonstration in collaborative environments using multiple experts (Castelli and Berman, 2008).

2.2.5. Summarization

Intelligent user interfaces often provide users with a wealth of information resources at their finger tips. This leads to the challenge of presenting growing amounts of information to users who have limited focus and time resources. One approach is to intelligently summarize the information into more manageable forms. Summarization often takes two forms: excision, which removes portions of the information, leaving only the most important parts to promote speedy information absorption by users, and highlighting, which includes all information but highlights the most important parts to preserve information context (Tucker and Whittaker, 2009).

Summarization-based approaches are grounded in earlier work studying text summarization of documents in the information retrieval domain, e.g. (Barzilay and Elhadad, 1997; Goldstein et al., 1999). However, as a growing number of information sources are no longer just text-based, e.g. audio/video sources, new techniques and applications of summarization have been recently proposed. For example, Basu *et al.* (2008) have applied summarization to audio logs of speech conversations and created an interface for presenting summaries at various levels of abstraction, navigable by users. Other recent advancements include using eye-tracking hardware to improve predictions of user focus for determining the most important pieces of information presented to users for future summarization (Xu et al., 2009).

Summarization techniques can also be used to create rich, context-aware interfaces to intelligently support users. For example, Wagner *et al.* (2009) created a news client which not only displays current events to users, but also extracts the most important parts of the news article (as in summarization) to know what types of information a user might want to investigate from the article. It then automatically discovers information from relevant sources, e.g. Wikipedia, and provides links to related content, allowing users to both further investigate the article's main ideas, as well as discover background information relevant to understanding the context of the news event. Similar interfaces could be developed for other domains, e.g. scientific investigation and discovery, with similar techniques.

2.2.6. Collaboration

As networking increases amongst humans and tasks become more distributed, requiring users of various skills and expertise, collaboration has also increased between users. Thus, computer scientists studying intelligent user interfaces have also conducted research on how to use artificial intelligence to support collaboration amongst distributed users.

In computer supported, collaborative work systems, users collaborate within work groups to accomplish shared tasks and achieve group goals. Recent research to support CSCW from the IUI perspective has included helping cooperating users form common ground, e.g. shared beliefs (Convertino et al., 2008), providing cues to experts that novice workers require assistance (Wong et al., 2007), providing simulated environments for collaborative scientific discovery in a classroom environment (Moher et al., 2008), and studying the communication chains formed between multitasking users (Su and Mark, 2008). The IUI community has also provided frameworks for building collaborative interfaces, such as the CRAFT project at IBM (Hupfer et al., 2009). Finally, intelligent user interfaces have also been used to improve collaborations in social networking. For example, intelligent user interfaces have provided technologies such as interfaces for growing social networks based on links between users mined from various information sources (Guy et al., 2009).

3. Adaptive Knowledge Assistants

In this section, we introduce the Adaptive Knowledge Assistants (AKA) framework, designed to provide personalized support to knowledge workers through an intelligent user interface. First, we describe the challenges to such personalized support, followed by the individual components which comprise AKA agents and a description of how these components mitigate the challenges of intelligent support. Afterwards, we relate the AKA components to the literature review provided in Section 2 to demonstrate how the reviewed IUI technologies are used to improve our agent components. Finally, we give a few examples of possible applications for which the AKA framework can provide intelligent support for human users.

3.1. Components

In the context of supporting knowledge workers, such as those conducting scientific investigations through cyberinfrastructures such as SCID, we recognize six important challenges to both user activities and providing adaptive, intelligent support: 1) difficulties in discovering knowledge from large quantities of raw data, e.g. not knowing how to form appropriate queries, not knowing what data to analyze versus ignore, 2) users have insufficient resources, e.g. time, skills, and patience, to complete all assigned tasks, 3) different users have different levels of expertise in the domain, 4) managing complex tasks, 5) coordinating collaboration between users, and 6) unobtrusively assisting users without increasing user frustration and decreasing productivity.

To mitigate these challenges, we have designed a framework where intelligent software agents called Adaptive Knowledge Assistants (AKA) provide customized support tailored to the human users of the knowledge system. Building upon existing research in intelligent user interfaces and multiagent systems, we have designed personal support agents acting in an intelligent user interface composed of five key components: 1) user modeling, 2) task management, 3) information management, 4) group support, and 5) user training. A summary of the actions provided by each component to support knowledge workers is provided in Table 1.

Table 1: Actions Provided by AKA Components

User Modeling	Task Management	Information Management	Group Support	User Training
• trace queries • track interface actions • elicit preferences • model neighborhood • determine expertise	• learn workflows • automate tasks • monitor task execution • schedule tasks • provide task reminders	• maintain information history • find related information • automated mining • maintain ontologies • summarize information	• matchmaking • user/research promotion • offline presence	• train from expert behavior • find experts for remote training

First, the user modeling component of AKA agents is responsible for observing, recording, and generalizing user behavior to identify patterns of repeated actions and characteristics (including querying behavior and interface interactions), elicit preferences about resources, tasks, etc. from users, determine the expertise of the user, and model the user's neighborhood of collaboration partners. These models of both the supported user and her partners are then used to guide the reasoning process of the agent to support its user.

Second, the task management component of AKA agents draws from prior work in personal assistants, e.g. (Myers et al., 2007; Conley and Carpenter, 2007; Yorke-Smith et al., 2009), to learn user workflows to support intelligent task interruptions, task planning, and automated execution. AKA agents also monitor the execution of both online tasks and batch tasks run offline while the user is away, schedule tasks and access to any required resources on the user's behalf, and provide reminders of tasks that need to be executed in the near future. Here, task management alleviates some of the burdens of keeping track of task executions and schedules by human users. Learning workflow models is also useful for possible workflow optimization based on domain expertise provided in the agent's internal knowledge base, further improving user productivity.

Third, the information management component of AKA agents operates similarly to the task management component, but instead of managing tasks, it organizes and retrieves information for users. Here, AKA agents maintain a history of information discovered and observed by users, automatically find information related to the current context of a user's activities, e.g. performing a task or investigating the results of a query, and automatically mine query results and other information for unknown relationships in the data and other important results to present to users. AKA agents also maintain semantic ontologies on information for users to federate information from different domains and sources, providing a “big picture” view, i.e. shared context, of the knowledge contained in the system. Finally, AKA agents can also summarize information for users, preventing user overload from wading through too much data.

Fourth, to support not only its assigned user but also the group activities the user participates in, AKA personal assistant agents are responsible for matchmaking users: finding collaborative partners based on the users’ tasks, expertise, and information needs. In a similar fashion, AKA agents can also promote the work of their individual agent to other agents and users, trying to advance the user’s agenda. This latter is especially important in a research setting (such as SCID) to promote the ideas of the user and works hand-in-hand with matchmaking to form collaborative groups of users with similar ideas and interests. Finally, AKA agents can also provide an offline presence for users when they are not available, keeping track of requests and messages from other users and possibly responding with the user’s permission.

Finally, AKA agents can also provide online training to improve the productivity and competency of their users. This training is based both on the observed behavior of experts within the system, such as training users to perform tasks better or how to use queries to investigate a data source. Additionally, agents can also find experts for more hands-on, human-human training through communication channels provided by the system, e.g. audio/video chat.

Together, these components handle the six problems inherent in intelligent support of knowledge systems as follows. First, the problem of extracting knowledge from large amounts of data is primarily handled by the information management component of AKA agents through automated information discovery and mining, as well as summarization. Second, resource insufficiencies are mitigated by task management, e.g. scheduling tasks and workflow automation, as well as through group support, e.g. finding experts to perform tasks. The problem of differing expertise levels amongst users is dealt with through user modeling for determining which users have different types of expertise, along with group support for forming groups with the required expertise to successfully complete joint tasks, as well as user training to improve expertise. Managing the complexities of tasks is mostly mitigated through task management activities, such as learning workflows and automated execution, while organizing team coordination is handled through group support. Finally, the problem of unobtrusiveness is embedded within the various modules, such as intelligently eliciting preferences from users during modeling, as well as intelligent notification for task monitoring.

It is important to point out here that this latter problem, along with other agent-human interactions, is handled by AKA agents in a *resource-aware* fashion cutting across all components, where agents model user “good will” as a valuable resource depleted by ill-timed interruptions or bad recommendations but restored through positive interactions with the system. The goal of this approach is to optimize the consumption of this limited resource to balance user frustration and stress with the urgency of interruptions and the behavior of the agent.

3.2. Relationship to IUI Research

To guide and organize our design of the Adaptive Knowledge Assistants, we used the literature review presented in Section 2 to find related work to both utilize and improve from the intelligent user interfaces community. Here, we describe the relationship between the various AKA components and the IUI research areas previously reviewed. A summary of this comparison can be found in Table 2.

Table 2: Relationship between AKA Components and IUI Research

	User Modeling	Task Management	Information Management	Group Support	User Training
Personal Assistant	model assigned user	manage to do list, task learning and automation	organize display, automated mining/retrieval	promote user, matchmaking	
Recommendations	use models in recommendations	recommend support tasks, resources	recommend queries, results	recommend partners, tasks	recommend learning tasks, resources
Intelligent Notification/Interruption	interruptions for eliciting personal information	monitoring, when to interrupt tasks	notify of key results	determine when to suggest collaboration	interrupt for training/help
Preference Elicitation	build models	preferred tasks	preferred displays, areas of expertise	preferred partners	preferred training methods
Learning from Humans	learn how users model each other	workflow learning and automated execution		learn how to form groups	learn how to train users from observation
Summarization	summarize users to each other		highlight results		
Collaboration	matchmaking, multiple viewpoints of user	find users to perform tasks	notify information held by users	group formation and support	find experts to help train

First, research on personal assistants obviously played a key role in the initial development of the AKA framework. Prior research on task management was central to the design of the related component within our framework e.g. (Myers et al., 2007; Conley and Carpenter, 2007; Yorke-Smith et al., 2009). However, task management is only single component of our personal assistant design. Here, our framework differs from others in that we also focus on information management, group support, and user training. We also differ from other personal assistant agents in the agent's reasoning process: as previously mentioned, we take a resource-aware perspective for guiding the interactions between user and agent.

Second, recommendations research was influential in the types of support provided by agents to users. For the various components, we need to recommend various "items", such as tasks to execute, queries to perform, users to work with, etc. As recommendations are not a key focus of our research, we don't aim to extend the literature but instead take advantage of the work performed by others previously.

Third, intelligent notification/interruption research plays an integral role in the relationship between users and agents. As we are considering a resource-aware approach to agent-human interactions, our agents' reasoning differs from previous approaches considered in the literature, providing an avenue for novel research. However, important results from the literature will be utilized in our agent's behavior, such as selecting notification representations based on the utility of the interruption (Obermayer and Nugent, 2000; Gluck et al. 2007).

Since our agents require user models to inform their reasoning process, previous work on preference elicitation will also be used to guide the modeling behavior of AKA agents, such as methods for extracting preferences from users, as well as internal representations for preferences. Furthermore, since preference elicitation is closely related to the user interruption problem as some preferences can only be obtained by asking users for direct feedback, we are in position to conduct novel research on balancing when to elicit preferences against the need for such preferences based on our resource-aware reasoning framework.

Next, since we want our personal assistant agents to be adaptive, one form of learning available is from human users. Specifically, our agents will perform programming by demonstration when learning tasks from observing user-originated task execution, as recently studied by Gervasio and Murdock (2009) and Yorke-Smith *et al*. (2009). We also want our agents to be able to learn other abilities from users, such as improving user modeling based on observing how humans model one another (if such information is accessible to agents), as well as learning how to form groups based on human collaboration activities. Furthermore, as related to the bootstrapped learning project (Oblinger, 2006), we are also interested in studying how agents can learn from instruction. However, rather than from a human user, we would like to investigate how well agents can learn by instructing one another, e.g. how to perform a task.

In relation to summarization, we again won't be conducting original research but can utilize the previous approaches developed by other researchers. Specifically, we are interested in using summarization in information management to highlight key (tasks and query) results, as well as summarize large collections of results. We also want to draw upon the context-aware aggregation of information sources proposed by Wagner *et al.* (2009).

Finally, as our AKA agents provide support for group activities, we also take advantage of previous research on supporting collaborations with intelligent user interfaces. Here, we are interested in group formation and support, including matchmaking users based on expertise or interest, as well as finding experts to help users train.

3.3. Application Overview

Given the numerous possible applications of knowledge systems, we see multiple opportunities for utilizing the Adaptive Knowledge Assistants framework to support human users. A sample of these opportunities include: 1) intelligent tutoring systems, where students interact with an intelligent system to learn concepts and generate new knowledge, 2) computer supported, cooperative learning (CSCL) or work (CSCW) environments, where users collaborate to accomplish tasks and generate shared knowledge, 3) and cyberinfrastructures for combining various knowledge and information sources for scientific discovery and investigation. To show the broad applicability of our framework, consider that specialized versions can also potentially be applied to intelligently support humans in other applications, such as smart environment controls, where the environment models human preferences, automats tasks such as light dimming or thermostat control, manages information about the environment from embedded sensors, and accounts for group preferences.

4. SCID and Biofinity

In this section, we introduce the Semantic Cyberinfrastructure for Investigation and Discovery, along with its first application: the Biofinity project (http://biofinity.unl.edu). We focus on the use of intelligence within the Biofinity interface, followed by a discussion describing how the Adaptive

Knowledge Assistant framework can be applied to handle the intelligent user interface needs of SCID/Biofinity.

4.1. Overview

The primary mission of the Semantic Cyberinfrastructure for Investigation and Discovery (SCID) is to provide an infrastructure for bridging information contained in distributed data sources from multiple scientific disciplines to support domain driven scientific investigation amongst collaborating scientists. To accomplish this mission, the SCID project has established three primary goals:

1) Unite scientific disciplines by federating distributed databases,
2) Make use of ontologies for relating concepts across information sources to support investigation and discovery, and
3) Streamline the use of scientific tools by accessing information sources through the relevant ontologies by way of a common, intelligent user interface.

First, by federating databases from multiple scientific disciplines, SCID offers users multiple perspectives on the same scientific data through the use of shared *context*. Using ontologies to map related concepts within these federated databases, SCID aims to unlock this context by building an organic knowledge base used to understand the information stored in its databases from both domain dependent and joint viewpoints. For example, in a biology application, one database might contain information about the physical characteristics of a collected specimen from the biodiversity perspective. However, another database might contain information about the same specimen at the genetic level from the genomics perspective. While each of these databases contain valuable information used to describe the specimen, together they offer a broader viewpoint for understanding 1) the internal processes within the specimen, e.g. how has its DNA caused the specimen to grow over its lifetime?; 2) the relationship between the specimen and its environment, e.g. how have pollution factors in the region influenced the mutation of the specimen's DNA over time; and 3) the trends in species change in a physical location over time, e.g. how might the terrain and flora of the location influenced changes in the species' genetic makeup over the last 50 years? Using different domain dependent tools, these questions might be answered by querying an ontology relating the information in the physical characteristics and genetics databases. These queries are made possible by an intelligent user interface offered by the SCID project, providing a common portal for access to the knowledge base created by its ontologies.

Currently, we are creating the first application of the SCID cyberinfrastructure: the Biofinity project (http://biofinity.unl.edu). Under this project, we are working to build an ontology relating the biology disciplines of biodiversity and genomics to answer questions such as those posited above. This project is focused on four objectives:

1) Creating a "seed" ontology relating the biodiversity and genomics data to start the knowledge base of shared context,
2) Develop an open Application Programming Interface (API) allowing access to the ontology from bioinformatics tools,
3) Produce collaborative editing tools to promote scientific discussion about new and existing results, and
4) Provide an intelligent user interface supporting ontology development, querying scientific data, and offering decision support to expert and novice users.

We have presently started the development of both the seed ontology and API and linked the two through the use of the first embedded application: MapViewer, a GIS tool for plotting collected specimen data for various species. We are now working on designing a semantic Wiki to serve as the collaborative editing tool, introducing data from external databases such as GBIF (http://www.gbif.net) into the ontology, and providing access to the ontology for other tools, including BLAST (Altschul, 1990).

4.2. Intelligent Support

In order to assist users with utilizing the SCID infrastructure and the Biofinity project for scientific investigation and discovery, we aim to use artificial intelligence to create an intelligent user interface to take advantage of the technologies outlined in Section 2 to improve both user productivity and the end-user experience. Specifically, we are focusing on three primary IUI activities: 1) ontology revision, 2) tracing user queries, and 3) managing user workflows.

First, in order to build an ontology relating various distributed databases across multiple scientific disciplines, our system must understand the semantic relationships between different concepts. By comparing the structure and content of two databases, some of this information can be automatically extracted, but manual input is still required from users, such as acquiring high level details for ontology refinement. However, rather than requiring users to always hand edit the ontology for refinement, our intelligent interface will ask users questions to refine its ontological rules and relationships, such as asking whether or not a link exists between two concepts the system believes should be related but with a low confidence, or requesting advice for resolving conflicts. Additionally, the intelligent interface could provide suggestions for improvements to the ontology which it can then act upon once advised by the human user, such as merging two existing ontologies. Over time, we aim for our interface to also be capable of learning from human demonstration to improve the automatic generation of ontologies from federated databases, requiring less future input from humans.

Second, whenever a user performs a query, information about the query structure, its parameters, and its results will be logged in our system, along with any relevant contextual information, e.g. what tools the user is currently working with. Over time, this will create a database of queries from multiple users with different skills and foci, useful for recommending queries to users based on their current task, as well as for training novice users based on expert behavior. This logged information might also provide clues about relationships between different information contained in the databases which might not be readily obvious but can be mined from user queries, improving both the scientific investigation process and ontology refinement. Tracked querying behavior also provides valuable information to the interface for building models of user behavior for further customization and personalized support.

Finally, we aim to use an intelligent user interface to manage user tasks through workflows. By tracking a user's behavior, we want to build models of recurring tasks for future automation. These models will also be useful for recommending resources for the user's task, such as which database to query or what tool to use for preprocessing or analyzing data.

4.3. Application of AKA Framework

We can now describe how to apply the Adaptive Knowledge Agents framework from Section 3 to the intelligent user interface activities within the SCID cyberinfrastructure as embodied in the Biofinity project. A summary of the support provided by the AKA components to each area of intelligent support desired by SCID/Biofinity is provided in Table 3.

Table 3: Relationship between AKA Components and SCID Intelligent User Interface

	User Modeling	Task Management	Information Management	Group Support	User Training
Ontology Refinement	model user expertise for ontologies	recommend tasks to find relationships in ontology	update ontology	find experts for refinement	train to improve refinement
Tracing Queries	model user query behavior, knowledge	recommend queries for tasks	track query history, find related information	find experts for information, model other users' queries	train to improve queries
Workflow Management	model user task behavior, expertise	learn workflows, automate tasks,	manage results of workflows	find task collaborators	train to perform tasks

First, the ontology refinement and query tracing activities are mainly information oriented, so they are primarily handled by the information management component of AKA agents. Here, actions such as maintaining information histories and ontologies, as well as automatically mining relationships from data can all be used to support automatic ontology refinement and query tracing. Modeling what questions to ask to improve ontologies and suggesting ontology edits can also be formulated based on information collected and organized within this module. The workflow management activity, on the other hand, is much more closely related to the task management component of AKA agents. With this component, AKA agents can perform such activities as workflow learning and automated execution.

While these two components primarily handle the desired intelligent user activities within SCID/Biofinity, it is worth noting that the other components of AKA agents can also provide support for these activities. For example, user training can be used to improve user expertise and competency in each of the activities. Likewise, user modeling provides information to the system for knowing how to reason about supporting these activities based on a user's behavior and characteristics. Finally, the group support component enables AKA agents to reason about these activities from a collaboration perspective, possibly incorporating more than one user in an activity at once.

It is also important to note that while the AKA framework is capable of providing the intelligent support desired by the IUI in SCID/Biofinity, it also offers many opportunities for additional support. For example, using AKA agents would also enable SCID/Biofinity to promote individual user's research to other research groups and users within the system, fostering group collaboration and data sharing – one of the primary goals of the SCID cyberinfrastructure. AKA agents could also intelligent summarize query results and automatically find important related information, improving users' ability to see both the "big picture" of the retrieved information, as well as provide suggestions for more in-depth analysis.

5. Conclusions

In this paper, we have presented a novel framework for supporting human users with knowledge generation tasks through the use of intelligent user interfaces and personal assistant agents. We also outlined the application of the Adaptive Knowledge Assistants framework to a real knowledge system in order to support scientific investigation and discovery within SCID/Biofinity. Additionally, we provided a brief literature review highlighting recent trends in IUI research related to our own and placed our framework and its application within the context of prior work.

In the near future, we plan to further flesh out the details of applying our framework to the SCID/Biofinity project, grounded in an actual implementation. This system will then be used to support actual knowledge workers with their scientific investigation of biology-related research. We will also expand on our review of existing IUI literature, especially as it relates to our core ideas and approach. Finally, we will extend our framework to include more activities for supporting knowledge workers, such as studying the possibility of creating intelligent visualization of query results, adapted to user preferences and information content.

Our immediate goals for studying the usefulness of applying the AKA framework to support human knowledge workers within the SCID/Biofinity project include:

1) Create a sandbox intelligent user interface, mimicking the semantic Wiki to be later implemented for SCID/Biofinity, to begin testing the implementation of AKA agents,
2) design and implement a strategy for modeling both general and task-specific user behavior, such as querying databases and executing tasks, especially focusing on the use of abstract-oriented programming (AOP) to fit within the overall software design for Biofinity,
3) experiment with *context-awareness* based on the queries posited by users, allowing the agent to find and present relevant supplementary information without overburdening the user, and
4) further develop our notion of *resource-aware* reasoning, especially as it applies to intelligently supporting users by balancing interruptions and notifications with user cognitive effort and frustration.

References

G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, June 2005.

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, Basic local alignment search tool, *Journal of Molecular Biology*, vol. 215, pp. 403-410, 1990.

D. Avrahami, J. Fogarty, and S.E. Hudson, Biases in human estimation of interruptability: Effects and implications for practice, *Proc. of CHI'07*, San Jose, CA, pp. 51-60, April 28-May 3, 2007.

R. Barzilay and M. Elhadad, Using lexical chains for text summarization, *Proc. of ACL Workshop on Intelligent Scalable Text Summarization*, 1997.

S. Basu, S. Gupta, M. Mahajan, P. Nguyen, and J.C. Platt, Scalable summaries of spoken conversations, *Proc. of IUI'08*, Canary Islands, Spain, pp. 267-275, Jan. 13-16, 2008.

G. Bonnin, A. Brun, and A. Boyer, A low-order Markov model integrating long-distance histories for collaborative recommender systems, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 57-66, Feb. 8-11, 2009.

C. Boutilier, A POMDP formulation of preference elicitation problems, *Proc. of AAAI'02*, pp. 239-246, 2002.

D. Braziunas and C. Boutilier, Elicitation of factored utilities, *AI Magazine*, 29(4), Winter 2008. pp. 79-92.

V. Castelli and L. Bergman, Distributed augmentation-based learning: A learning algorithm for distributed collaborative programming-by-demonstration, *Proc. of IUI'07*, Honolulu, Hawaii, pp. 160-169, Jan. 28-31, 2007.

H. Chalupsky, et al., Electric Elves: Applying agent technology to support human organizations, *Proc. of IAAI '01*, Seattle, WA, Aug. 7-9, 2001. pp. 51-58.

J-H. Chen and D.S. Weld, Recovering from errors during programming by demonstration, *Proc. of IUI'08*, Canary Islands, Spain, pp. 159-168, Jan. 13-16, 2008.

L. Chen and P. Pu, Survey of preference elicitation methods, Human Computer Interaction Group, Ecole Politechnique Federale de Lausanne, EPFL Technical Report IC/2004/67, 2004.

D. Cosley, D. Frankowski, L. Terveen, and J. Riedl, SuggestBot: Using intelligent task routing to help people find work in Wikipedia, *Proc. of IUI'07*, Honolulu, Hawaii, pp. 32-41, Jan. 28-31, 2007.

G. Convertino, H.M. Mentis, M.B. Rosson, J.M. Carroll, A. Slavkovic, and C.H. Ganoe, Articulating common ground in cooperative work: Content and process, *Proc. of CHI'08*, Florance, Italy, pp. 1637-1646, April 5-10, 2008.

A. Cypher, ed., *What what I do: programming by demonstration*, MIT Press: Cambridge, MA, http://acypher.com/wwid/, 1993.

J.A. Fails and D.R. Olson, Jr., Interactive machine learning, *Proc. of IUI'03*, Miami, Florida, pp. 39-45, Jan. 12-15, 2003.

M.T. Gervasio and J.L. Murdock, What were you thinking? Filling in missing dataflow through interference in learning from demonstration, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 157-166, Feb. 8-11, 2009.

J. Gluck, A. Bunt, and J. McGrenere, Matching attentional draw with utility in interruption, *Proc. of CHI'07*, San Jose, CA, pp. 41-50, April 28-May 3, 2007.

J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, Summarizing text documents: Sentence selection and evaluation metrics, *Proc. of SIGIR'99*, Berkeley, CA, pp. 121-128, 1999.

I. Guy, I. Ronen, and E. Wilcox, Do you know? Recommending people to invite into your social network, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 77-86, Feb. 8-11, 2009.

Y. Hijikata, T. Shimizu, and S. Nishida, Discovery-oriented collaborative filtering for improving user satisfaction, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 67-76, Feb. 8-11, 2009.

E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Roomelse, The Lumiere Project: Bayesian user modeling for inferring the goals and needs of software users, *Proc. of UAI'98*, Madison, WI, pp. 256-265, July 1998.

P-Y. Hsueh and J.D. Moore, Improving meeting summarization by focusing on user needs: A task-oriented evaluation, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 17-26, Feb. 8-11, 2009.

R. Hu and P. Pu, A comparative user study on rating vs. personality quiz based preference elicitation methods, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 367-371, Feb. 8-11, 2009.

B. Hui, G. Partridge, and C. Boutilier, A probabilistic mental model for estimating disruption, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 287-296, Feb. 8-11, 2009.

S.C. Hupfer, et. al, Crafting an environment for collaborative reasoning, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 379-382, Feb. 8-11, 2009.

T. Kulesza, et al., Fixing the program my computer learned: Barriers for end users, challenges for the machine, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 187-196, Feb. 8-11, 2009.

P. Maes, Agents that reduce work and information overload, *Communications of the ACM*, vol. 37, no. 7, pp. 30-40, 1994.

R. Mailler, D. Bryce, J. Shen, and C. O'Reilly, MABLE: A framework for learning from natural instruction, *Proc. of AAMAS'09*, Budapest, Hungary, pp. 393-400, May 13-15, 2009.

T. Moher, B. Uphoff, D. Bhatt, B.L. Silva, and P. Malcolm, WallCology: Designing interaction affordances for learner engagement in authentic science inquiry, *Proc. of CHI'08*, Florence, Italy, pp. 163-172, April 5-10, 2008.

K.L. Myers, et al., An intelligent personal assistant for task and time management, *AI Magazine*, 28(2), pp. 47-61, Summer 2007.

J. Nichols and T. Lau, Mobilization by demonstration: Using traces to re-author existing web sites, *Proc. of IUI'08*, Canary Islands, Spain, pp. 159-168, Jan. 13-16, 2008.

R.W. Obermayer and W.A. Nugent, Human-computer interaction for alert warning and attention allocation systems of the multi-modal watchstation, *Proc. of SPIE'00*, vol. 4126, pp. 14-22, 2000.

D. Oblinger, Bootstrapped learning – external materials, DARPA, http://www.sainc.com/bl-extmat/, 2006.

J. Orkin and D. Roy, Automatic learning and generation of social behavior from collective human gameplay, *Proc. of AAMAS'09*, Budapest, Hungary, pp. 385-392, May 13-15, 2009.

B. Peintner, P. Viappiani, and N. Yorke-Smith, Preferences in interactive systems: technical challenges and case studies. *AI Magazine*, 29(4), pp. 13-24, Winter 2008.

P. Pu and L. Chen, User-involved preference elicitation for product search and recommender systems, *AI Magazine*, 29(4),pp. 93-103, Winter 2008.

A. Ritter and S. Basu, Learning to generalize for complex selection tasks, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 167-176, Feb. 8-11, 2009.

J. Shen, et. al, Automatically finding and recommending resources to support knowledge workers' activities, *Proc. of IUI'08*, Canary Islands, Spain, pp. 207-216, Jan. 13-16, 2008.

J. Shen, E. Fitzhenry, and T.G. Dietterich, Discovering frequent work procedures from resource connections, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 277-285, Feb. 8-11, 2009.

N.M. Su and G. Mark, Communication chains and multitasking, *Proc. of CHI'08*, Florence, Italy, pp. 83-92, April 5-10, 2008.

M. Tambe, E. Bowring, J.P. Pearce, P. Varakantham, P. Scerri, and D. V. Pynadath, Electric Elves: What went wrong and why, *Proc. of AAAI Spring 2006 Symposium*, 2006.

R. Tuchinda, P. Szekely, and C.A. Knoblock, Building data integration queries by demonstration, *Proc. of IUI'07*, Honolulu, Hawaii, pp. 170-179, Jan. 28-31, 2007.

S. Tucker and S. Whittaker, Have a say over what you see: Evaluating interactive compression techniques, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 37-46, Feb. 8-11, 2009.

J. Vig, S. Sen, and J. Riedl, Tagsplanations: Explaining recommendations using tags, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 47-56, Feb. 8-11, 2009.

E.J. Wagner, J. Liu, L. Birnbaum, and K.D. Forbus, Rich interfaces for reading news on the web, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 27-36, Feb. 8-11, 2009.

J. Wong, L.M. Oh, J. Ou, C.P. Rose, J. Yang, and S.R. Fussell, Sharing a single expert among multiple partners, *Proc. of CHI'07*, San Jose, CA, pp. 261-270, April 28-May 3, 2007.

S. Xu, H. Jiang, and F.C.M. Lau, User-oriented document summarization through vision-based eye-tracking, *Proc. of IUI'09*, Sanibel Island, Florida, pp. 7-16, Feb. 8-11, 2009.

N. Yorke-Smith, S. Saddati, K.L. Meyers, and D.N. Morley, Like an intuitive and courteous butler: a proactive personal agent for task management, *Proc. of AAMAS'09*, Budapest, Hungary, pp. 337-344, May 13-15, 2009.