calo

final report

mhci project course, spring
may 16, 2007

SRI Team CALO

Yoko Nakano, Will Haines,
Margaret Szeto, Jing Tien, Brian Ellis

Contents

1 Executive Summary

The Defense Advanced Research Projects Agency (DARPA) is responsible for all the research and development projects undertaken by the U.S. Department of Defense. A handful of project managers oversee all of these projects, each responsible for the progress and continued existence of his or her respective projects. In carrying out these responsibilities, project managers are inundated by meetings, presentations, important emails, and many other methods of acquiring and disseminating information. Not all of these tasks require the unique expertise of a DARPA project manager, yet they are frequently difficult to prioritize.

The CALO project (Cognitive Assistant that Learns and Organizes) is intended to permit project managers and other "overburdened knowledge workers" to offload responsibilities that are candidates for automation to an autonomous software agent. In the spring of 2007, SRI International, the primary center of development on the CALO project, contracted the Carnegie Mellon University Human-Computer Interaction Institute to suggest an improved model of interaction for CALO and to research novel uses for the system that have not heretofore been explored.

The CALO team at Carnegie Mellon is a multidisciplinary group, consisting of human-computer interaction students with backgrounds in design, psychology, and computer science. Over the past several months, the team has been exploring the existing literature and conducting a detailed user analysis to define both the problem space and the available solutions, with the intent of developing a prototype that supports an improved model for CALO's interaction with end users. The following report presents the research and user analysis conducted by the team, and the insights gleaned from this process.

Over the course of the research phase of our design process, the CALO team discovered a number of interesting findings. Our review of current literature pertinent to CALO identified several key areas of interest: human collaboration, information management, agents, and multimodal interfaces. We found that while many technologies exist to support collaboration and information management, often they are overloaded with uses that differ from their original intended design. Systems already exist to handle collaboration and interruption management, but they are either disruptive or too specialized to benefit CALO without further research. Our review of current cognitive agents indicates that we must take social factors, especially autonomy, into account when designing our desired mental model for CALO. Finally, we note that multimodal interactions have been successfully leveraged to reduce the extrinsic costs of interacting with agents and improve support of collaboration.

Our second phase of research, user studies provided a number of key insights that will inform our design for CALO. Service professionals, both assistant and executive, face difficulties in task prioritization in the face of constant

interruption and tasks that require waiting on others. In light of this, we intend to leverage CALO's knowledge and learning to support time management, provide non-invasive support to changing user priorities, reduce cognitive load on end users, and act as a buffer between the executive's desired work style and that of the outside world. This model of interaction with CALO reflects the work flows observed in our interviews with executives and assistants, aiming to augment, rather than replace, these collaborations.

2 Literature Review

CALO is a highly complex system, the design of which draws upon many well-researched topics in computer science, human-computer interaction, and other fields. Our research goal was to familiarize ourselves with these topics and determine what impact each research area has had on the development of CALO up to this point, and how this research is likely to continue to affect it in the future.

Based on our exploration of the CALO publications and those of related projects, the four themes that appear to have most directly impacted CALO's development can be grouped into collaboration, information management, agents, and multimodal interfaces. The applicability of research into agents is certainly not in doubt given the nature of CALO, and that collaboration and information management should be a major factor is evidenced by the fact that many of CALO's major subsystems (such as the meeting assistant, PTIME, and Towel) are all information management tasks geared at least in part toward facilitating collaboration between CALO users. Multimodal interfaces are not particularly integrated into CALO at this point in time, but they are associated with context-aware computing and software agents in the literature for some time, and there seems to be considerable interest on the part of the developers in making multimodality an increasingly important part of CALO's user interaction. For this reason, we considered them to warrant further study as well.

2.1 Collaboration

In our review of literature pertaining to CALO, we first focused on the ways people collaborate with each other and the ways that software (especially agents) can support this interaction. As early as 1990, Grosz and Sidner [22] theorized that successful collaboration stems from mutual understanding about the goals, action, capabilities, intentions and commitment of the participants who collectively form a shared plan. However, Barthelmess et al. [4] observed that current collaborative technology is disruptive and does not support natural human to human communication. Rather than a series of user-command and system-display turns, they suggest a system with several unobtrusive sensors that recognize and process interaction in the background while creating appropriate artifacts. Multiple modalities are explored to construct a system that proactively identifies user's intentions.

One large area of interest in collaboration with human users is the realm of intelligent interruption management. From a theoretical angle, Iqbal and Bailey [25] examined the feasibility of building statistical models that can detect and differentiate three granularities of perceptually meaningful breakpoints during task execution, without having to recognize the underlying tasks for determining the breakpoints for optimum interruption. Fogarty et al. [20] used a variety of sensors to improve an agent's ability to interrupt human users at more appropriate times. In an experiment with users performing a programming task, the sensor-based system was able to determine interruptability correctly 72% of the time, and the research discussed what sort of sensors are most useful for inferring context in programming tasks. Such a system may be useful in CALO, but more research would be necessary to determine appropriate sensors for context-awareness in non-programming situations.

Avrahami and Hudson [2] found that IM conversations offer several benefits for users, including the ability to selectively attend to or ignore messages, but are typically highly interruptive and do not allow users to easily prioritize important questions and information above less important messages. They successfully implemented an IM plug-in to determine whether incoming messages were likely to be important, and choose whether to interrupt the user depending on this determination. Adamczyk and Bailey [1] devised an interruption management system for monitoring and specifying user tasks using physiological measures of workload and task modeling techniques in order to systematically and automatically identify opportune moments in a user's task sequence to mitigate the negative consequences of interruptions by notifications.

Another large area of collaboration research is in ways to support cooperation via email, a heavily overburdened system. Dabbish, Kraut, Fussell, and Kiesler [16] proposed a model of email use to predict whether or not a given email will be replied to, with the some modeled factors including using inbox visibility for reminders, keeping information requests and responses in the inbox, and responding to but not filing meeting requests. Using action requests, status updates, reminders, information requests and responses, scheduling requests and responses, and social content as proposed email types, the authors were able to use a regression model to match email importance to likelihood of response. They found that content, job complexity, and sender characteristics are good indicators of response, and that the identity of the responder plays a larger than expected role in predicting filing due the tendency of people to either sort or search their email but not both. Again leveraging email's multiple uses, Shen, Li, Dietterich, and Herlocker [44] described TaskPredictor, an application that uses Naive Bayes classification and confidence thresholds to try to guess what task a user is currently performing based on what system resources are currently being used. Tests on a corpus of email showed an promising 80% classification accuracy that could possibly be improved through more computationally expensive methods such as Hidden Markov Models.

A final area of collaboration support is in the realm of agents to help schedule and document human meetings. Faulring and Myers [19] presented Rhaical, an intelligent calendaring system that proposes natural language support and novel visualizations to help users when scheduling meetings for multiple parties. The agent might contact users to verify assumptions, get confirmation, and allow the user to understand and control its behavior through natural language processing and manipulation of a calendar visualization. For actual meeting documentation, Ehlen, Niekrasz, and Purver [18] described the CALO's Meeting Assistant. This assistant analyzes multi-party speech and handwritten input, identifying the meeting's topics and action items, and displaying a high-level summary report (along with the user's own manual notes) in a browser, so as to compel users to make manual corrections, and to suggest action item transfers to other agents such as the Towel to do manager. The combination of user feedback, integration of analyzed input and manual notes, and collaboration with other agents produces a highly personalized representation that parallels the user's perception of the salient aspects of the meeting.

2.2 *Information Management*

Information management, including the organization and retrieval of information, is an increasingly complex problem as the amount of data users encounter continues to grow. Several tools have been developed to support users and their data such as email, to do lists, and calendars, but these applications are often overloaded and have weak boundaries. To address this problem, information management has been moving from manual tasks done by the user to more agent-based applications for automating processes. Several parameters have been studied, such as the hierarchy knowledge workers employ to organize their information that would better support their workflow in the contexts of email and file folders. Research also spans on other applications that are designed to convey information, send reminders, handle task management and regulate scheduling.

Boardman and Sasse [12] studied information management across tools, specifically files, email and Web bookmarks and long term issues relating to personal information management. They found that the nature of acquisition varied between tools from manually done in files and bookmarks to uncontrolled in email. File management strategies also varied from file on creation to file on completion of task or during a "spring cleaning." Similar patterns were found in email management where they found no filers who do not organize and instead search their email, frequent filers who file as emails come in and spring cleaners who file their email from out of their inbox at intervals.

Martin and Jose [38] reveal the other software that facilitates information management includes information retrieval system prototypes such as Fetch, which adopts the concept within an information-seeking environment specifically designed to provide users with the means to better describe a problem they don't understand. Along with Fetch, another piece of software, created by Bao et al. [3], FolderPredictor, works in the same problem space. FolderPredictor applies machine learning algorithms to the observation of users' opening and saving of files, analysis of document content, and the making of context-aware predictions to reduce the amount of time users spend locating their files.

Henderson [23] looked at the attributes knowledge workers use to structure their information into hierarchies. Genre, task, course, topic, time, and person were the most frequently used folder types. Some of the dimensions like person, source, topic, time and file type can be automatically supported by software whereas genre, course/task and security are unsupported by software automation and must be done manually. Lerman, Gazen, Minton, and Knoblock [34] used automated grammar generation, a automated technique to group data into hierarchies, to semantically mark up data-filled websites and tag them based on heuristics. This method labeled automobile sales data columns correctly 64% of the time, but inconsistent data formats and similarly formatted but semantically unrelated fields remain as hurdles to greater accuracy.

Other parts of information management include the use of email. An interesting example is the U.S. Government investigation into the Enron collapse which resulted in a large corpus of email messages analyzed by Klimt and Yang [31]. They found that while most users make use of folders to organize their email, it is also important to classify messages by thread and relationship to other messages, a difficult problem for a computer if human users fail to use 'reply' to maintain thread relationships. They also discovered trends in the data that indicate useful ways of classifying messages using the message body and from fields.

Besides organizing information, applications like email take on multiple roles of information management. Bellotti, Ducheneaut, Howard and Smith [5] recognized the transition of email as a task management tool supporting to dos, ongoing correspondence, delegation and receiving of work. To address the growing complexity of email, they created Taskmaster, an email system designed for task and project management. In their research they identified the following seven problems and designed Taskmaster accordingly. Taskmaster works by "keeping track many concurrent actions (the user's and the ones expected from others), making important things salient amongst less important items, managing activity over time (keeping track of threads of activity and discussions), managing deadlines and reminders which can be associated with other content, collating related items and associated files and links, application switching and window management, and getting a task oriented overview rather than a glance through scrolling or inspecting folders.”

Some similar functionality already exists in CALO. Its front end, IRIS, is the interface that integrates all the different components of CALO’s intelligent agent system for the user to operate. Cheyer, Park, and Giuli [13] summarized the concept of semantic desktops, intelligent knowledge management and systems for augmenting the performance of human teams and how IRIS was designed with components borrowed from existing semantic desktops and knowledge management software.

Conley and Carpenter [15] presented Towel, an intelligent to do list manager developed under CALO, is another tool somewhat similar to Taskmaster that handles task management with direct communication with the user. A style of digital communication between the user and Towel is vital to Towel's operation and training, and instant messaging’s model of interruptions (opening a chat window and playing sounds), status information for different contacts, contact list (strikingly similar to the look of a to do list), and flexibility to carry out either rapid human-to-human dialogs or lax conversations (such as hiding the chat window until a more appropriate time) provides an ideal framework for the workings of a to do list. Also, the chat windows limit the number of operations the user can undertake, and they also make users directly manipulate an operation using commands, so to avoid dealing with anything outside of the context of the task.

As for action items that are verified and managed by Towel, they reach the user again in the form of notifications and reminders. In *Effective Interaction Strategies for Adaptive Reminding*, Weber and Pollack [41] discuss that a robust reminding system should consist of a motivating justification, attention to reminder granularity, user's preferred signal, and machine learning techniques. There are two general approaches that these learning techniques have taken: one called reinforcement learning where the machine refines its reminding algorithm based on a cumulative reward system, and the other called supervised learning where the machine selects and presents certain data to the user for training.

Other applications that also share this management space include calendars. Modi et al. discuss [39] the CMRadar calendar management component is capable of making autonomous scheduling decisions, negotiating schedules with other users and agents, and prioritizing existing meetings to determine how to resolve scheduling conflicts. CALO’s calendar component, PTIME, is an agent-based scheduler that learns, as Berry et al. [9] discussed in their paper *A Personalized Calendar Assistant.* Some features include the ability to work with the user to solve infeasible scheduling problems, automated preference learning and automatic inferences about when best to interrupt the user, backed by active, procedural, and especially reinforcement learning techniques. Later, Berry et al. [8] discusses the PTIME system is organized around the principle that people dislike giving up control over their schedules, whether to a software agent or otherwise. Since users often have widely differing preferences and practices in regards to time management, PTIME is designed to support and augment, rather than replace, the user's natural processes. Berry, Myers, Uribe, and Yorke-Smith [6] built this scheduling agent based on soft constraint-solving, allowing the system to autonomously create goals and reason about user commitments. They suggest that good constraint-based scheduling algorithms to handle scheduling already exist, but inherent uncertainty in user schedules requires even more robust responses to dynamic schedule requirements if a satisfactory system is to be fashioned.

Both of CALO's scheduling assistants, PTIME and Pisces, take the collaborative approach, called Mixed-Initiative, of balancing scheduling algorithms and human evaluation of schedule quality and nuances of domain constraints. PTIME is designed to learn and refine the user's preference model, whereas Pisces is more focused on providing solution to very large and complex problems. Berry et al. [7] expressed the hope that the scheduler's autonomy will grow with time, and indicated that reinforcement learning may be the best candidate to make this hope a reality.

2.3 *Agents*

With the expansion in amount of information people deal with on a daily basis and the advance of AI technology, computer agents are increasingly being incorporated in user interfaces. The notion of a cognitive agent that dynamically accommodate to user's workflow cannot be made possible without AI agent components.

Mike Papazoglou [40] summarized four different types of agents: application, general business activity, information brokering, and personal agents. Other types of agents generally fall under these main types. Application agents are application specific agents that are specialized to a single area of expertise and work cooperatively with other agents to solve a complex problem in the domain. A procurement agent is an example of an application agent. General business activity agents take care of typical commerce transactions such as business purchasing, billing, parsing information on the Web, and finding trading partners. Information brokering agents (also referred to as matching agent) "maintain, update, and access distributed directory services," as well as performing advanced navigation services. Brokering agents help service distributors publish their services and customers to look for these services. Personal agents work for specific users and their needs "to support the presentation, organization and management of user profile, requests, and information collections" distributed on the Web and the personal computers. Personal agents need to monitor and learn user habits and activities and may suggest better ways of performing these tasks. Examples of personal agents are intelligent tutoring systems and Web browsing assistants.

All computer agent designs must be designed with theoretical considerations and practical concerns to be successful. Through her study on interaction between users and cooperative AI agent that can initiate communication, monitor events and perform tasks, Maes [36] raised important issues related to topics such as agent personification, mental models, styles of training, privacy of users, and the responsibilities of agent's actions and transactions.

Kaye and Karam [30] presented a design for distributed cooperating knowledge based assistants that emulate the behavior of human office assistants. These assistants cooperate with each other to complete tasks initiated by the user and interact with conventional office systems such as databases and message systems. The purpose of the agents is to relieve office workers from having to learn and use a large variety of systems or having to integrate tools to build high level applications.

Rich and Sidner [42] stated that autonomous agents should be governed by the same principles that underlie human collaboration and communication during shared tasks. Computer agents have varying degrees of autonomy determined by the granularity of the task and user's needs. Usually, the user of the agent decides how much of a task to delegate to the agent. Alternatively, multiple-agent systems might identify different components of a task and delegate them to other agents inside the system. In a more interesting case, Maheswaran, Tambe, Varakantham, and Myers [37] discussed the concept of adjustable autonomy—the ability of an agent to decide when to cede control to a human user or to ask for confirmation. Schurr, Varakantham, Bowring, Tambe, and Grosz [43] examined the adaptation of Isaac Asimov's laws of robotics to teams of autonomous or semi-autonomous agents. They found that, perhaps contrary to expectation, rigidly following human orders at all times leads to degradation in agent team performance and an increased, not decreased, likelihood of bringing harm to humans. This effect can be mitigated by communicating the agents' misgivings with orders so that users can suggest alternatives. More research needs to be done to understand miscoordination costs among groups of human users and agents or in situations with uncertain knowledge states.

As for agent training, computer agents could possibly be trained in a similar manner as human assistants to human assistants. Agents can be trained explicitly, by observation and imitation, and by receiving positive and negative feedback from the user. The challenge is for agents to learn correct sets of information and provide enough feedback to the user so that they could un-train incorrect assumption. Tambe et al. [45] discussed a set of semi-autonomous personal software agents termed "electric elves" placed in a work environment resulted in increased efficiency but also in several large social and workflow breakdowns when the user was unable to correct the agents' faulty assumptions.

Kozierok and Maes [32] point out that both memory-based learning and reinforcement learning approaches would allow users to build up trust with the agent as it learns the user's habits, making suggestions and predictions, coupled with explanations and confidence levels for the user to verify. Empirical testing indicates that the user-agent pair is more effective at the given task than a pair of human users.

Garera and Rudnicky [21] discussed an agent designed to help users create weekly summary documents by making inferences from raw data as opposed to finished text. Despite this difficulty, a system trained on hand-classified data helped users complete the summary task in 22% less time over the course of the study. Unfortunately, automatic classification was less precise, leading the authors to suggest direct instruction, information synthesis, and active information acquisition as future supplements to improve the system. Tomasic, Zimmerman, and Simmons [46] aimed to create an agent that could help users find and fill in forms in complex corporate knowledge bases. Using natural language processing, and user-agent feedback loop system, the agent was able to retrieve the correct mini-form 80% of the time, an acceptable rate given the difficultly that humans have with this task.

To do lists prove to be a challenge to intelligent user interfaces, as presented by Gil and Chklovski [14] in terms of having to map users' natural utterances to internal task representations, anticipate minor and preparatory tasks to accomplish users' tasks, to determine the context of the tasks and know its own limits, and to know when task automation is desirable. The structure of BEAM includes the syntactic parsing of the user's natural language in reference to several repositories of external and internal organization knowledge, and the collaboration with other agents (such as SPARK) within the CALO architecture in order to execute automated tasks.

The CMRadar agent presented an integrated component of Outlook and provided a novel interface for explaining its scheduling decisions to the user. It also established an interesting paradigm for multi-agent interaction—how agents communicate with each other entirely through emails. Modi and Veloso [39] also demonstrated multi-agent scheduling and rescheduling—how an agent takes into account the density of another user's schedule to access the difficulty of scheduling a meeting with that person.

With regard to transfer of knowledge inside a single or between multiple agents, Marx, Rosenstein, Kaelbling, and Dietterich [17] discuss that knowledge transfer is profoundly complicated because the decision boundaries for different tasks exist in different feature spaces. Through an experiment where they observed the processing of two separate tasks, they found that while a machine can find the model of the first task to predict the parameters of the second, this only happens accurately if the tasks were generated from a common source and existing in the same domain.

2.4 Multimodal Interfaces

Multimodal interfaces—that is, the use of multiple modalities such as speech, gestures, written commands, etc.—have long been considered prime candidates for the interfaces of agents such as CALO. Several authors have explored the various strategies and ramifications of providing multimodal input, as well as specific opportunities that lend themselves particularly well to multimodal interfaces.

Lunsford, Kaiser, Barthelmess, and Huang [35] described a set of "extrinsic costs" that are incurred when humans, who naturally interact multimodally, are constrained to unimodal computer interfaces. These include the need to over-specify or re-specify input to satisfy the computer, and the overhead of the interface misinterpreted user behavior. They also discussed ways that multimodal interfaces can reduce or eliminate these costs.

Huang and Oviatt [24] showed that multimodal input is often sequential rather than simultaneous, and that the choice of sequential or simultaneous input is very consistent within users. Some users were also observed to consistently choose unimodal methods of input even when multimodal input was available. Further, Krause, Siewiorek, Smailagic, and Farringdon [33] showed that physiological information such as stress level and movement patterns can be used to predict interruptability and determine the context of the user's interaction with a wearable computer. The authors also made the point that non-intrusiveness and minimal active training are both essential features of a successful context-aware system.

Kaiser [28] explores new methods to supplement speech recognition by combining it with handwriting analysis rather than lip-reading and relying on mutual disambiguation techniques to acquire out-of-vocabulary words. In baseline test, detection rate of new words was 100% with greatly improved error rate and accuracy metrics; however, the test data set was too small to make any definitive conclusions . Kaiser et al. [29] created Charter, a system developed to support remote collaboration. Charter used multimodal sketch recognition, vision based body-tracking, and speech/writing recognition for minimal intervention on work practices. In this system, the inputs can be displayed to distributed members in other locations. Charter can learn new terms used by the group and build semantic interpretations based on interaction .

Biehl and Bailey [11] studied comparing how well three classes of interfaces, textual, map, and iconic, support application management during realistic, collaborative activities in a multiple-device environment (MDE) and found that users preferred and performed better with the iconic interface due to its more comprehensive visual and spatial representation.

In another paper, Kaiser [26] discusses the SHACER (Speech and HAndwriting reCognizER) software's capabilities of learning new terms dynamically from single human-to-human interactions during multi-party meetings, applying knowledge of persist across related meetings, and determining the semantics of handwritten abbreviations. Lastly, Kaiser, Demirdjian, et al. [27] demonstrated the collaborative creation of Gantt scheduling chart using multimodal interfaces including gesture recognition, handwriting recognition, natural speech processing and body tracking.

2.5 Research Summary

From our research, we have identified several areas of key research interest relating to CALO and many insights from previous work in these areas. In particular, we see that current collaborative technology can be disruptive to the very collaboration it is meant to support, and that interruption management may play a key role in mitigating this effect. We also note that people tend to leverage existing technologies such as email and instant messaging and overload them to take on new responsibilities and tackle new tasks. Agents like CALO must take this into account and seek to leverage existing technologies like these itself.

We have also discussed several extant examples of cognitive agents, some of which have been deployed and observed in the field. Agents clearly fall into many different categories, all of which behave somewhat differently. The design and implementation of an agent must take into account social as well as technological factors, since agents often take an active role in their users' social environment. We also see several different approaches to human-agent interaction, with varying degrees of autonomy. We also observe differences in mental models and training styles, and varied approaches to the degree of personification expressed by the agent.

We note further that multimodal interaction is of central importance to CALO and other cognitive assistants; research indicates that multimodality can solve or reduce the impact of many of the problems we have discussed, by reducing or eliminating many of the extrinsic costs of interacting with a computerized agent. Multimodality also has direct relevance to collaboration, since a multimodal interface can integrate much more tightly into a highly collaborative setting with minimal intrusion.

Much of this research was an interesting exploration of how collaboration is managed in a professional environment and what an agent could be capable of, but it was still unclear to us how this would apply to our target user group in the context of their work. We therefore embarked on a series of user studies to supplement this research, which is described in the next section.

3 User Studies

3.1 *User Pool Justification*

To understand the needs of CALO's target user group, overburdened knowledge workers, we looked at two user types: assistants and executives. Executives fit the demographics of busy professionals who face the complexities of dealing with multiple projects and people at any given time. Assistants are a secondary user group identified because of their relationship with and importance to the primary target user group, executives. The assistant's job is also to focus on the workflow; therefore, they are better able to describe the mechanics of their work whereas executives focus on a high-level view and tend to disregard irrelevant details.

Finally, in order to understand the use of CALO by people with either substantial training or experience using the system, we obtained data from the CALO developers. Data from these individuals not only provided us valuable insights into how CALO is incorporated into users' actual work practices but also the perspectives from which different developers approached the problem domain.

After extensive focus setting sessions, we came up with two main areas of focus to direct our contextual inquiries: first, how do people collaborate on the job and what software supports this? And second, what mental model should CALO support to meet user needs?

3.2 *Contextual Inquiry Overview*

To collect data on our user groups, we conducted contextual inquires to obtain insights and breakdowns about their workflow. Contextual inquiry, as defined by Beyer and Holtzblatt [10], is a method in which the researcher goes to the user's workplace to learn and understand his or her work in the context in which it lives. After a contextual inquiry is conducted, the entire research group meets to create models of the data collected. There are 5 models: flow, cultural, sequence, artifact and physical, which are created to reflect different, but important parts of the user's work. The flow model captures the responsibilities and flow of information and artifacts associated with the user's job. The cultural model records influences that come from groups or organizations that the user perceives onto themselves. The sequence model documents the steps and procedures the user takes to accomplish his or her tasks. Artifact models are representations of actual documents or things that the user uses in his or her workflow. Lastly, the physical model is a map of the user's physical workspace to capture where the user works and its effect on the user's workflow. These models are created for each individual user and then consolidated by user type to gather insights about the user group rather than individuals and their details.

3.3 *Our Contextual Inquiries*

We conducted 14 contextual inquires over a three month period with our three user groups: assistants, executives and CALO developers. Within the 14 contextual inquiries, there was some overlap between executives who were also developers.

4 Consolidated User Models

After gathering a large amount of data at the granularity of a single CI user, we consolidated the models that we generated in order to visualize the data at the level of a user archetype. In this way, we are able to factor out the idiosyncrasies of individual users and design from more general trends that will support our user base as a whole. Since CALO must serve a number of very different users in different ways, we decided in this case that it would be most instructive to consolidate our models into three user archetypes—the developer, the administrative assistant, and the executive. These specific archetypes were motivated both by the groups of target users specified by SRI and our modeling process. Each archetype provides us with a varied set of insights and requirements for our design. They also served to motivate our final focus.

4.1 *SRI Developers*

Our trip to SRI's main campus in Menlo Park, CA, provided us with both a number of insights into the ways in which end users might interact with specific parts of CALO and a high-level overview of the ways in which the developers envision integration for CALO as a whole. However, as indicated on our developer flow model, their interaction with CALO was generally more limited and artificial than one would hope to see with an end user. Typically, a developer would focus on training and using the part of CALO that they were actively developing, more as a debugging procedure than as an actual user. As such, their interactions were more hypothetical than the sort of data typically observed in contextual inquiry.

While the data gathered was most instructive from a CALO-demonstration perspective, it was interesting to note that developers tended to struggle with the components of CALO that they were not actively developing. This indicates that in its current form, research CALO, the interface requires too much low-level knowledge to operate. By observing less technical users in the field, we hope to propose an improved model of interaction for end users.

such as purchasing or making travel arrangements. This knowledge makes them a resource to employees for whom they are not directly responsible.

Interestingly, the executives for whom the assistants are responsible seem to encourage this behavior, "lending out" their assistants to perform tasks for their clients and office mates (Fig. 3). The cultural tendency of many assistants to be unable to turn down requests for help also exacerbates this propensity, which at times leads to feelings of being overwhelmed. CALO probably cannot directly support outside responsibilities, but there exists an interesting parallel between lending out one's assistant and skill transfer by the CALO agent.

4.2.4 Desire for Perfect Knowledge

The insights discussed so far all relate to aspects of the assistant's experience that greatly increase their work load and level of stress. As such, assistants are typically highly overwhelmed, and develop coping strategies to deal with this. By far the most prevalent is to seek "perfect knowledge" of the work of which they are a part. We observe that assistants try to know everything that is transpiring in their realm of influence, whether or not it is useful or relevant to them at that moment, due to their perception that they are the "last line of defense" for those who depend upon them (Fig. 3). They perceive that if they fail to take the appropriate actions in response to any external event, no one else will be able to correct their mistake before it has dire consequences. This perception also motivates the assistant to double-check everything they themselves do, to ensure that nothing has slipped through the cracks. CALO, acting as a repository for organizational knowledge, can both support this desire explicitly and reduce the cognitive load on the assistant.

4.2.5 Trust Over Time

In general, the executive and assistant relationship is one of increasing trust and responsibility over time (Fig. 3). Assistants tend not to be explicitly trained, firstly because there is insufficient time, and second because it is not always clear what the assistant should be trained to do. Instead, we typically see an assistant's functions expanding organically over time with increasing autonomy for them to manage items such as their executive's schedule and travel arrangements. This relationship definitely ties to the concept of adjustable autonomy in CALO, and warrants further exploration.

4.2.6 Frequent Use of Databases

One notable difference between secretaries and coordinators is the tendency of coordinators to interact with databases on a regular basis. Therefore, this interaction likely results because coordinators are responsible for supporting a larger number of people than secretaries, and databases facilitate handling many employees. Breakdowns arise because many coordinators are not particularly technical, and they treat these databases as "black boxes." Further, assistants tend to duplicate effort when asked to input transfer paper data into the database (Fig. 2). CALO's task learning component would likely be useful in reducing the burden on coordinators interacting with databases.

4.2.7 Consistent Support

A final insight from our data on assistants is that while the executives they support are very different, assistants tend to support them in consistent ways. Some common activities are scheduling meetings, handling traveling arrangements, managing financial transactions, and providing reminders (Fig. 5). It may be most advantageous to design CALO to support assistant workflows because the applicability of such an approach would extend to a large number of fields, whereas designing for a specific type of executive has a lower generality.

4.3 Executives

4.3.1 Decentralized Information

Information applicable to the executive is typically spread across many different repositories, and it exists in many different forms (Fig. 6). Many executives view their assistants as useful for collecting and distilling all information into one form that is easily digestible. CALO seems well suited to collecting information from a diverse number of repositories, so it may prove useful to support visualization of this information.

4.3.2 Buffering Work Styles

The executives interviewed each had different preferred tools and styles of working. They expect their assistants to act as a buffer between their preferences and those of others with whom they interact. It is more important that assistants learn their executives' styles of work than their actual job description (Fig 7). In fact, we see executives desiring to bring their assistants with them to new jobs for exactly this reason. CALO has this portability—the ability to learn and maintain the executive's preferences is paramount. Learning to handle a large number of data formats is a foreseeable problem that would certainly need to be addressed at some point in the future.

4.3.3 Collaboration is Fundamental

As we interviewed higher level executives, we noticed that their jobs get more service-based. Typical work requirements include creating reports and presentations, setting requirements, and reading large amounts of email (Fig. 6). These sorts of activities require communication among parties who are often physically separated. CALO already has some facilities that support collaboration, such as meeting annotations and presentation generation, yet it can be better integrated to support executive workflows.

4.2 *Administrative Assistants*

During the consolidation process, we found that while the cultural, physical, sequence, and artifact models were consistent among all assistants, the workflow models differed to such a great extent as to imply the existence of two archetypes. The differences centered around whether the assistant was fully responsible for a small number of executives or was responsible in a more limited way for a larger number of lower-level employees. We named these archetypes the secretary and the coordinator, respectively. Once we made this distinction, we were able to draw a number of important insights from our completed models.

4.2.1 Constant Interruptions

Our first interesting discovery is the observation that while assistants are constantly interrupted, elimination of their interruptions is not a viable goal for CALO. Instead, we see that these interruptions are an integral part of a workflow that is based around serving a large number of people for relatively short amounts of time (Fig. 1). Thus, instead of reducing these interruptions, we should focus on ways in which to support sequences that are resilient to interruption.

4.2.2 Waiting for Others

Another workflow aspect that leads to a number of breakdowns is the necessity of waiting for external information. Commonly, this information comes from people, not databases, so the assistant is required to wait for the provider to actually get around to responding to their request (Fig. 2,3). The end result of this waiting is to fragment work sequences and cause the assistant to handle many tasks in parallel. This makes task prioritization difficult since it is not possible to simply follow one task through to completion.

Because of the difficulty of prioritizing tasks and a need for flexibility, the most common practice is to keep these tasks either on paper or simply in the mind, a set up that is prone to errors (Fig. 4). The problem is amplified for coordinators who have to deal with an even larger number of constituents who may be distributed across the office or further. Ideally, CALO will be able to serve as a repository for these sorts of short, pending tasks.

4.2.3 Extra Responsibilities

The next insight, taking on responsibilities outside of one's job description, appeared with almost every assistant interviewed. It seems counter-intuitive to think of going outside of one's job description as an intrinsic quality of being an assistant, but the reasons behind such a phenomenon are equally as interesting. Over the course of working, assistants gain knowledge in some specific domains

4.3.4 No Common Sequences

We recognized that supporting the nature of the executives' work is not as important as supporting the underlying communication between them and other parties. Any non-specific, repeated sequence represents an inefficiency in the executive's workflow because such repetitive tasks should be within the responsibility of their assistants. As such, the sorts of sequences that CALO would have to support for executives would necessarily be domain-specific. Thus, CALO would have to be designed with this domain knowledge in mind, although ultimately it may be possible to make CALO customizable by an expert user.

5 Focus

These insights, taken together, serve to inform and support the direction and focus of our design process. In particular, we see many opportunities based on our research for improving and augmenting the time management portion of CALO. Additionally, we intend to explore ways of bringing together all of CALO's knowledge and learning abilities to support the problems we have identified related to time management. By learning what users are doing at the moment, what they should be doing, and what they will likely be doing in the future, CALO can help users to prioritize the tasks they need to perform, keep track of tasks that may depend upon external factors, such as those that require waiting for other people. This approach will improve the adaptability of CALO's time management features while simultaneously reducing the cognitive load on end users. We also intend to explore ways of having CALO adapt to changing user priorities, and provide non-invasive support and suggestions. Lastly, we plan to search for methods of enhancing, rather than replacing, existing collaborations between executives and their assistants.

6 Preliminary Design

6.1 Personae

Personae are personifications of the user archetypes that our research identified. They exist to present characteristics of the user models in a form that is easier to think about and design for.

Janine

"I can't say no."
"I need to know everything."
"I'm old school."

About Janine

Female, 47 years-old
Lives in Chelsea, MA
Upper Middle Class
Makes $46,000/yr.
Drives a 2003 Toyota Camry

Job Description

Janine is an adminstrative assistant at Gaither & Associates, LLP, a medium-sized law firm in Boston, Massachusetts. Her job is to assist her boss in handling travel arrangements, arranging meetings, and handling purchasing. She goes into the office at 8:30am, and usually leaves around 6:30pm, or whenever she finishes all the work that her boss requires her to do for that day. She has worked for her boss for the past 4 years, and sometimes calls him "Bobby."

Life Story

Janine grew up in Atlanta, GA. She attended Agnes Scott College, a national liberal arts college for women. She received an associate's degree in English, and was planning on pursuing a bachelor's degree when she decided to quit school and get married. While she had some side retail jobs at Woolworth during her college years, she became a full-time housewife and mother for the next 15 years. When her two kids were just toddlers, her husband's job was relocated to Boston, so the entire family moved to Massachusetts. When her children entered high school, she decided to re-enter the work force and found a job as an assistant at a law firm. She is not entirely computer literate, but she had great organizational skills and eventually picked up the technical knowledge she needed for her job.

Richard

"I don't have time for anything irrelevant."
"I need to concentrate on my work."
"I can't leave until this is done."

About Richard

Male, 41 years-old
Lives in Chicago, IL
Upper Class
Makes $127,000/yr.
Drives a 2005 Mercedes E450

Job Description

Richard is a business consultant for Fantus, LLP, a consulting company that handles corporate site selections in Chicago, Illnois. He often travels to sites, visits client companies, accompanies his clients to the sites, and holds meetings with them and his colleagues. His schedule is often unpredictable, and it requirces him to stay in his office until his work is done. He relies heavily on his secretary to arrange his frequent travel.

Life Story

Richard grew up in Toledo, Ohio. He received his MBA degree at the University of Michigan, and his first job was being a assistant supply chain manager at Gillette in Cincinnati. He got married and had one daughter. He accepted a job offer from Fantus, so his entire family relocated to Chicago. After a few years, he and his wife filed for a divorce due to his pressures at work, and now he sees his daughter twice a month. Richard maintains a healthy lifestyle, on top of working 60+ hour weeks.

Sharon

"I can't say no."
"I wish I had enough time to help everyone."
"I can't think about my job linearly."

About Sharon

Female, 33 years-old
Lives in San Rafael, CA
Middle Class
Makes $36,000/yr.
Drives a 2004 Honda Accord

Job Description

Sharon is a coordinator at Lifehouse Incorporated, a charitable organization that helps people with developmental disabilities. She mainly coordinates payroll, supplies, and travel arrangements, especially when the organization sends employees out to attend conferences. She begins her day at 7:00am, and leaves strictly at 3:00pm so that she can pick up her son from school.

Life Story

Sharon grew up in Santa Cruz, CA, and attended the University of California, Berkeley, where she majored in Economics. At her first job, she worked at a bank as a customer service agent in San Rafael. She got married, and eventually quit her job when she gave birth to her son. She spent the next 5 years as a full-time mother, then picked up a part-time job at Lifehouse when her son entered pre-school. She started off as a receptionist, and after a couple of years, she became a full-time coordinator for the organization. Sharon is very devoted to her work, and also to her family. Sometimes she has to bring her son into the office with her, because there simply is not enough time for her to complete her work and take care of her family at the same time.

6.2 Use Cases

A use case is a general scenario of how a particular sort of user might use a software system. Use cases are valuable for their ability to reveal system requirements and user roles, and also for solidifying nebulous foci into well-defined system capabilities. Since collaboration is among our foci, all three types of users: secretaries, coordinators, and executives appear in our use case diagram (Fig. 8). Some of the use cases with which these users are involved and which fall within our focus area include scheduling and being reminded about meetings, adding to-do items and being notified about them, and interacting with CALO's perception of the user's priorities; all of these will be priorities for prototyping.

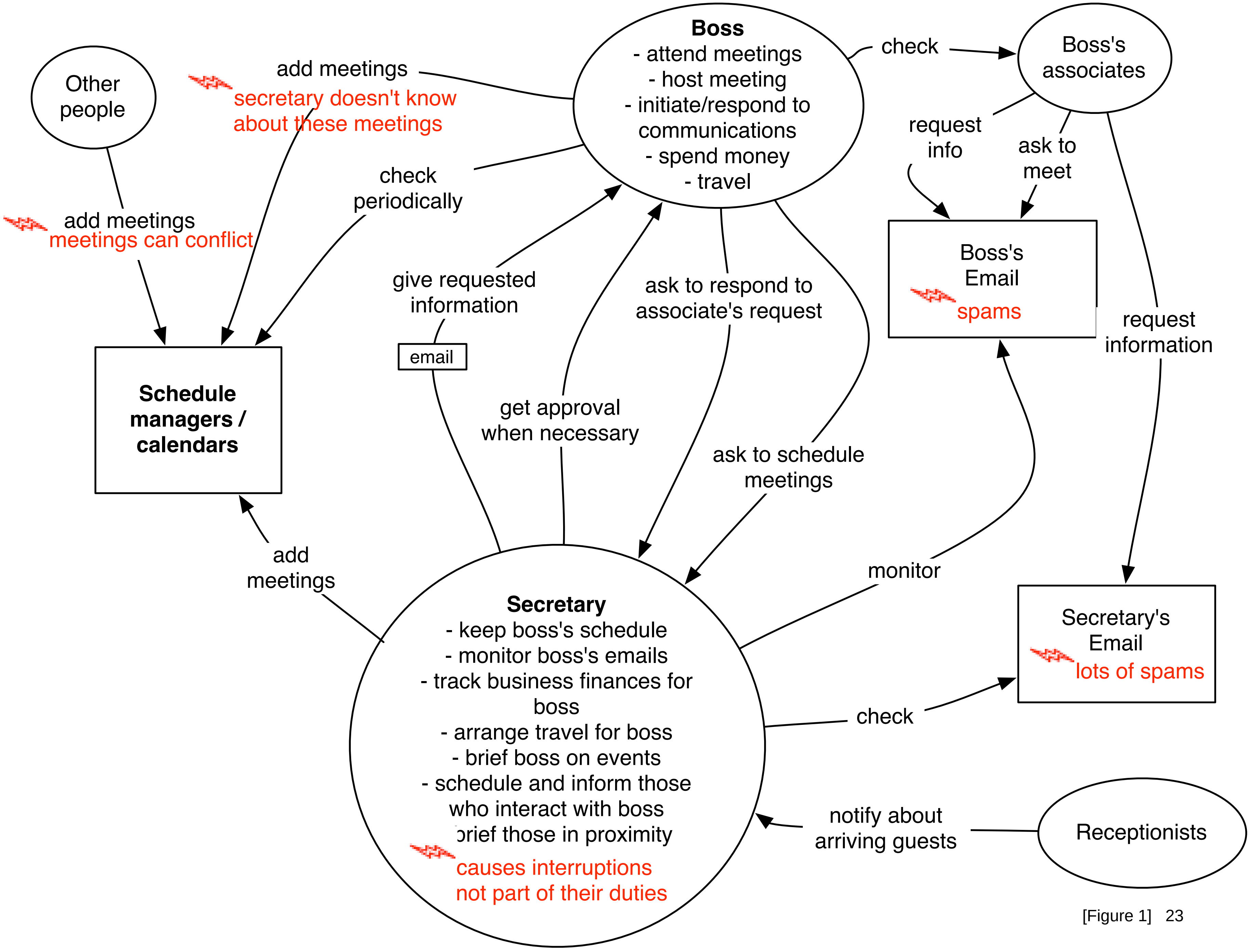
7 Future Directions

Our review of the existing research in the areas related to CALO, coupled with our contextual inquiries and consolidation have allowed us to establish a sound grasp on both the problem and solution space. Our findings from the consolidated models will guide us through our next phase: ideation. Through ideation, we will be creating prototypes and beginning an iterative process of user testing, utilizing HCI methods such as think aloud user studies. Ultimately, we plan to produce a high fidelity prototype that simulates CALO's functionality.

8 Appendix A: Consolidated Models

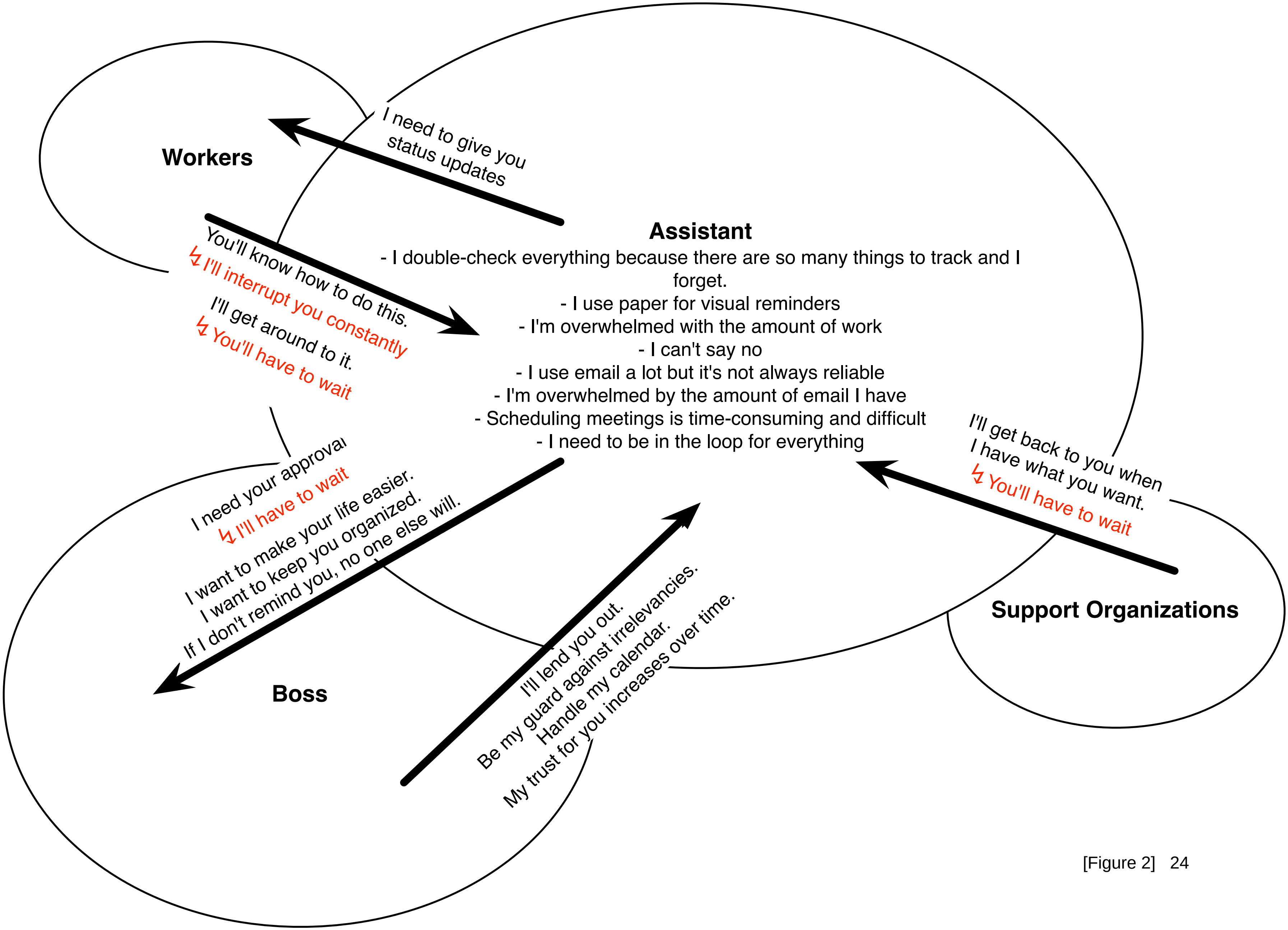
The models on the following pages were created when we consolidated the contextual inquiries described in sections 3 and 4. They contain a more complete characterization of the findings from the consolidation phase, and provide background for some of the summary statements we have made in the preceding pages.

Consolidated Secretary Flow



[Figure 1]

Consolidated Cultural Model — Assistant



Workers

I need to give you status updates

You'll know how to do this.

↯ I'll interrupt you constantly

I'll get around to it.

↯ You'll have to wait

Assistant

- I double-check everything because there are so many things to track and I forget.
- I use paper for visual reminders
- I'm overwhelmed with the amount of work
- I can't say no
- I use email a lot but it's not always reliable
- I'm overwhelmed by the amount of email I have
- Scheduling meetings is time-consuming and difficult
- I need to be in the loop for everything

I'll get back to you when I have what you want.

↯ You'll have to wait

Support Organizations

I need your approval

↯ I'll have to wait

I want to make your life easier.

I want to keep you organized.

If I don't remind you, no one else will.

Boss

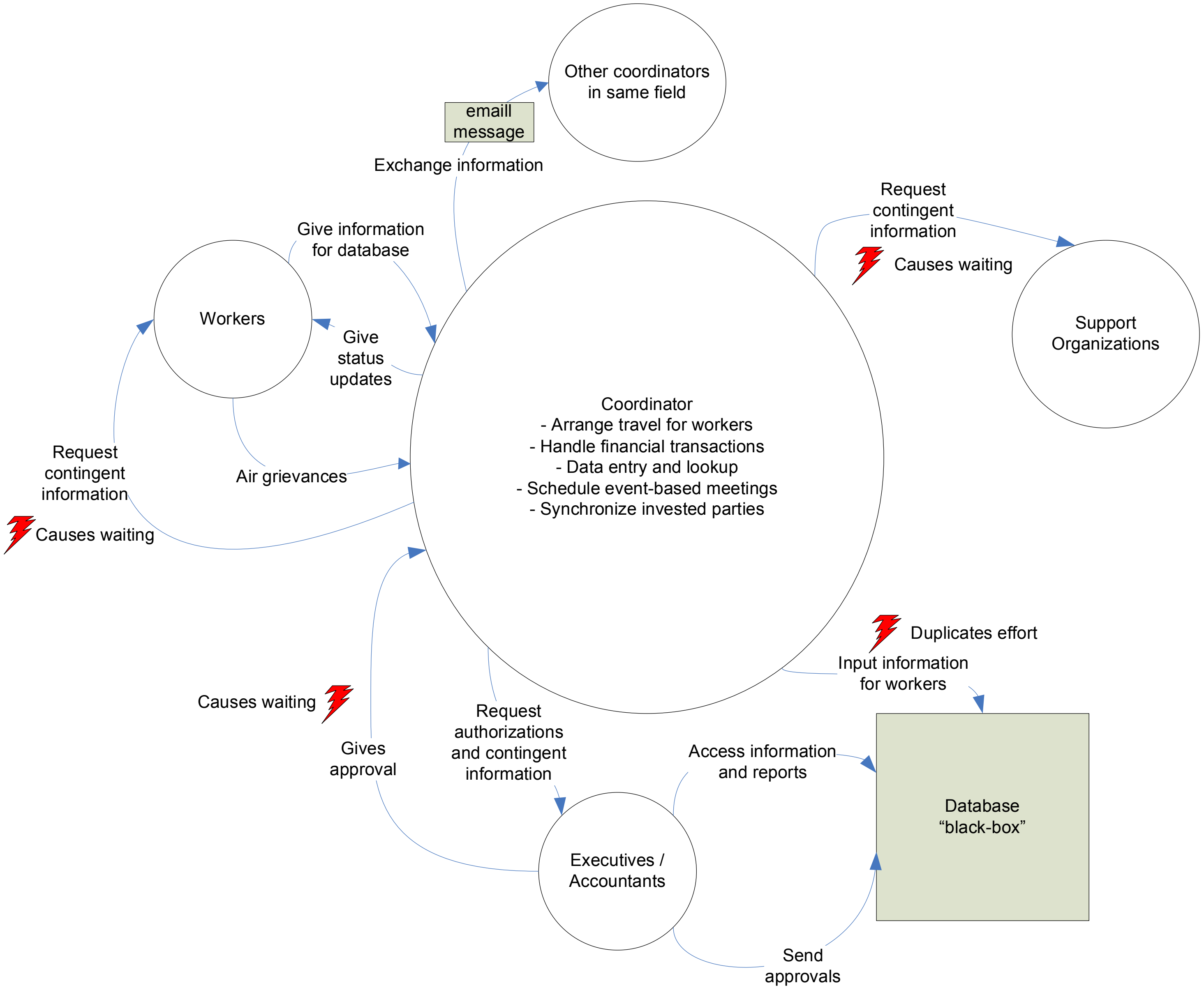
I'll lend you out.

Be my guard against irrelevancies.

Handle my calendar.

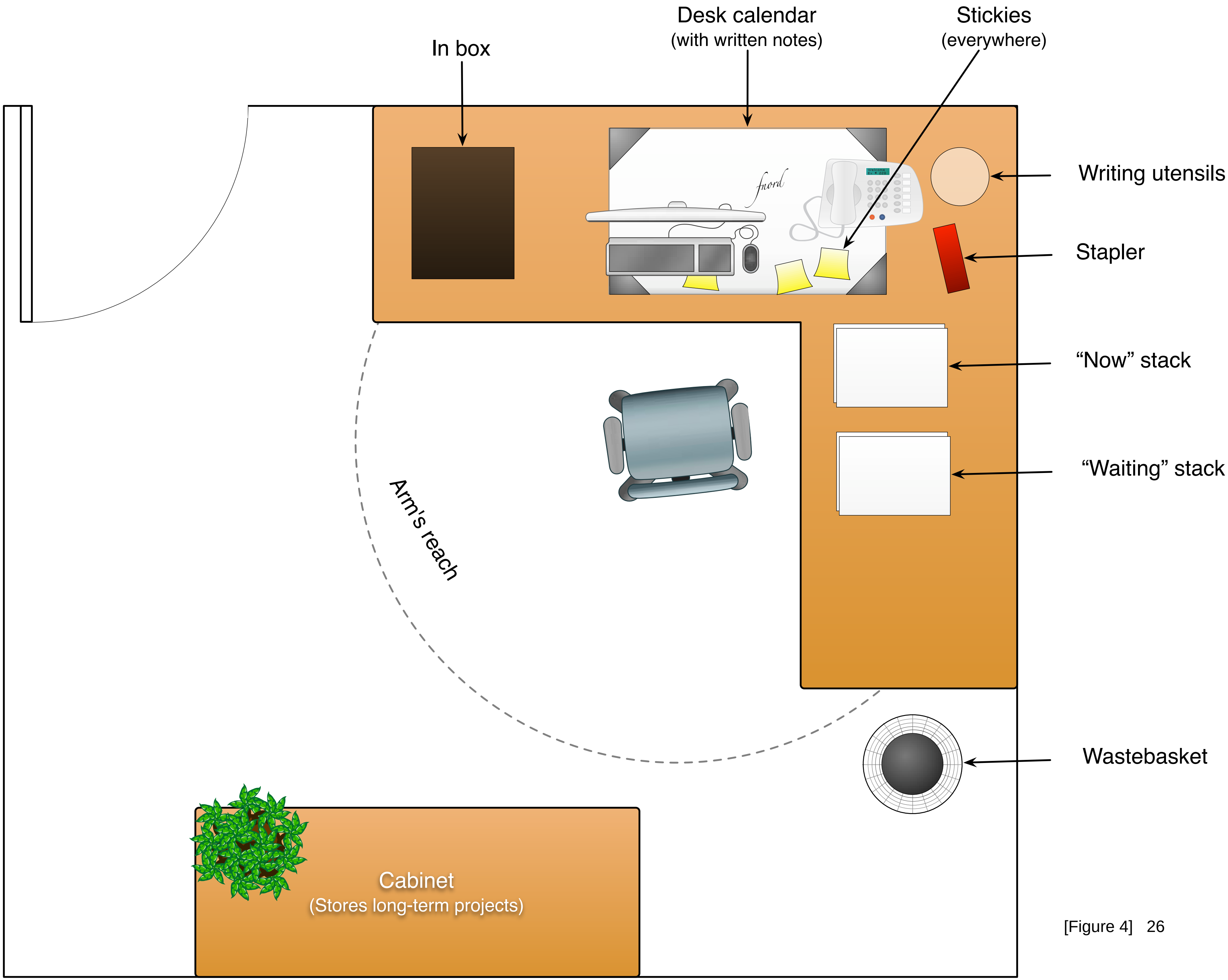
My trust for you increases over time.

Coordinator Consolidated Flow



[Figure 3]

Assistant Physical Model (Consolidated)



In box

Desk calendar
(with written notes)

Stickies
(everywhere)

Writing utensils

Stapler

“Now” stack

“Waiting” stack

Arm's reach

Wastebasket

Cabinet
(Stores long-term projects)

Consolidated Secretary/Coordinator model

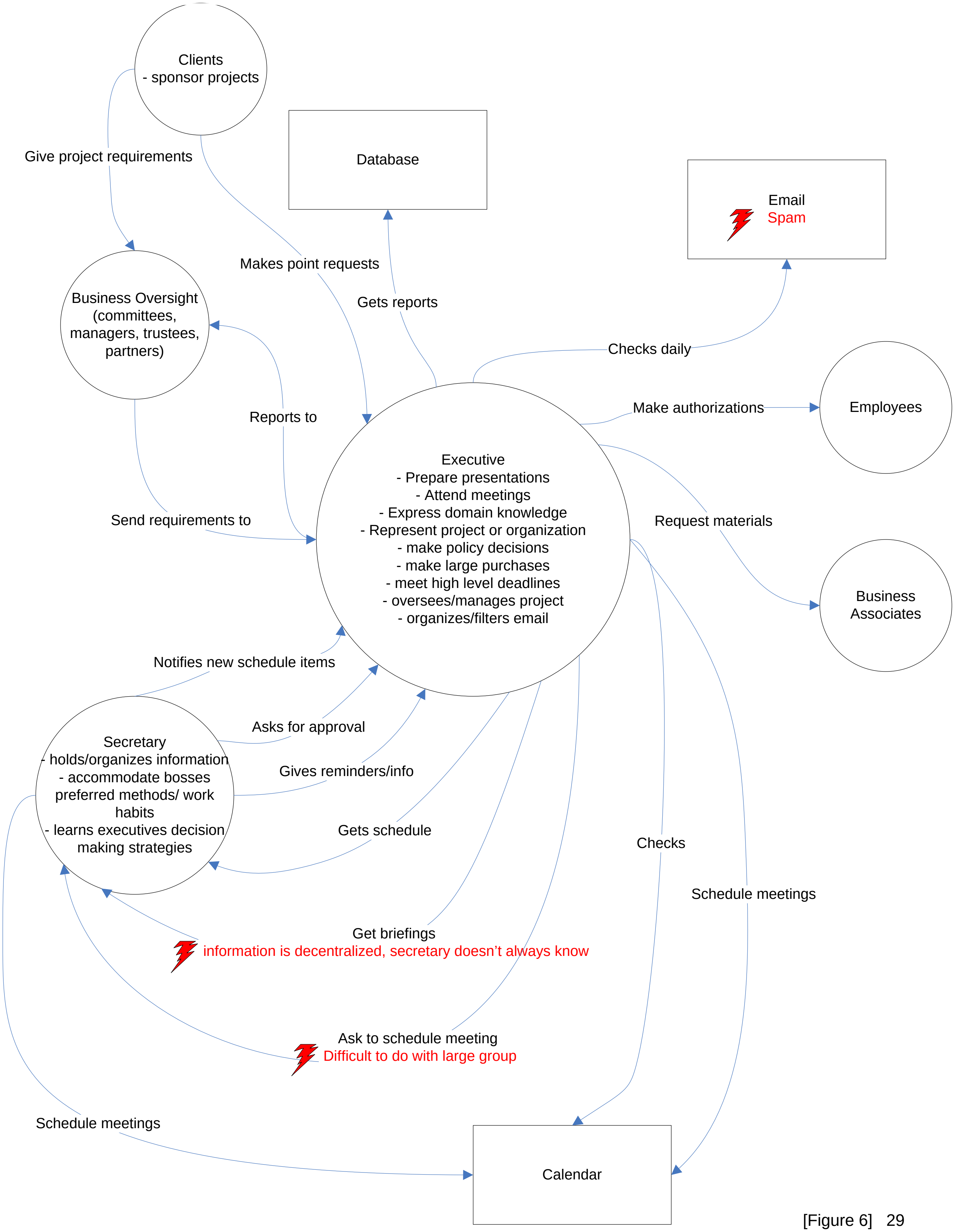
Activity	Intent	Abstract Step
Schedule Meeting		Trigger: gets request for meeting
	Determine meeting constraints ex. Time, date, people	Look up or email appropriate people, check boss' schedule
	Ensure venue availability	Check room scheduling database
	Ensure participant availability	Email participants or check schedule if available ⚡ have to wait for responses no centralized database out of date datebase
		Trigger: key participant is unavailable
	Cancel meeting	Email participants or remove from calendar if available
		Trigger: meeting impending
Make sure meeting happens	Remind about meeting	Email participants and walk into boss' office to remind them

Activity	Intent	Abstract Step
Determine travel possibilities		Trigger: someone needs to travel
	Determine travel constraints (when, how long, where)	Look up event information online communicate with traveler
	Gather information	Contact travel agent or travel websites
	Negociate with traveler	Email or present options to traveler
Schedule travel	Request funding	Fill out a form and submit to a database or accountant
	Readjust schedule	reschedule meetings if applicable

	Build itinerary	prepare/consolidate documentation and send to traveler
Travel wrap up	Manage reimbursements	Trigger: employee has receipts ⚡ if employee kept receipts Gather receipts, fill out forms and forward to database of accountant

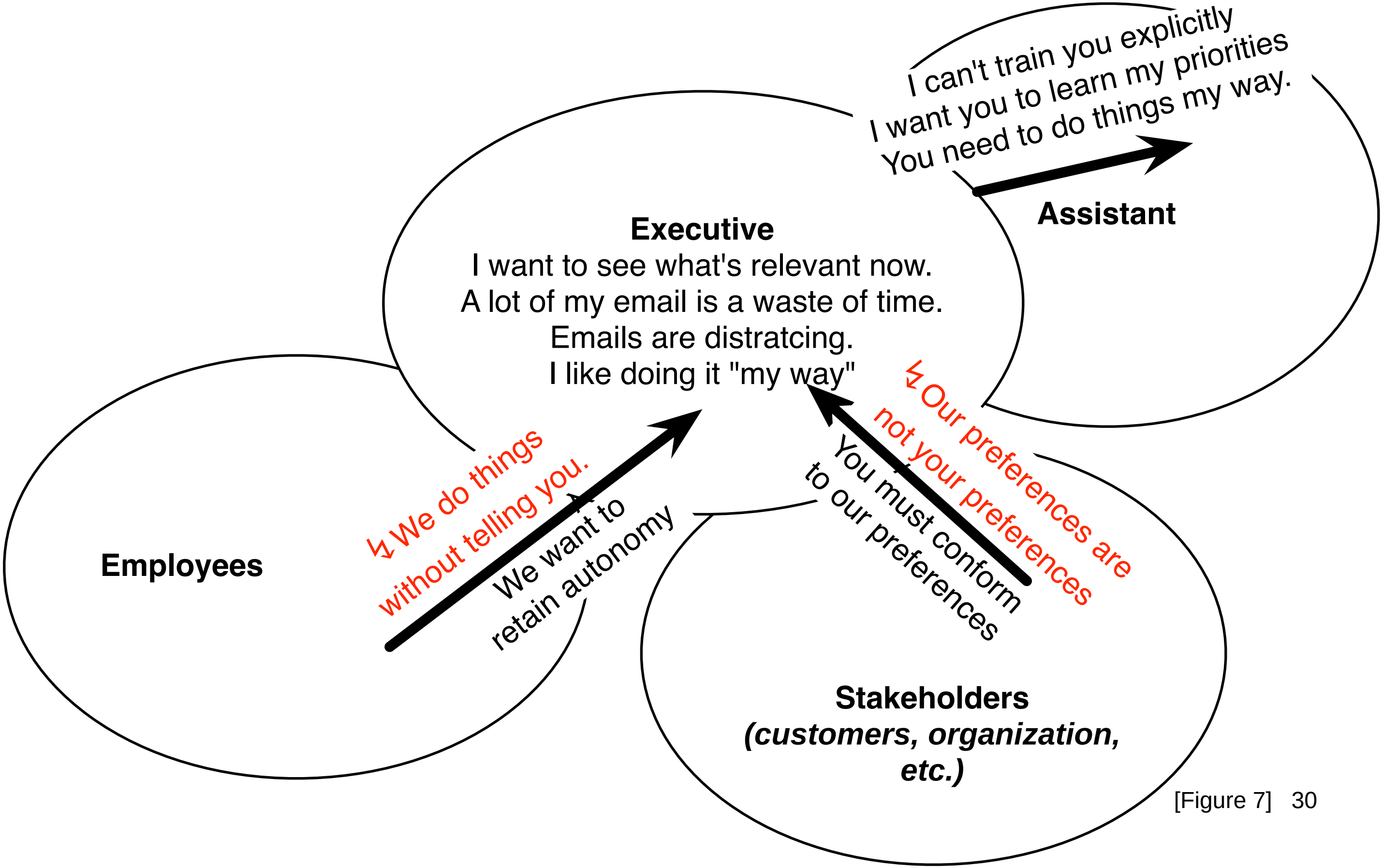
Activity	**Intent**	**Abstract Step**
Manage financial transaction	Collect information	Trigger: gets request for financial transaction (purchase, reimbursement, payment) Receive artifacts that justify the transaction ⚡ artifacts may be incomplete, cause further steps to get all information
	Request authorization	Submit a request for authorization through email or database
	Notify accountant	Enter into database

Consolidated Executive Flow model



[Figure 6]

Executive Consolidated Cultural Model



Executive
I want to see what's relevant now.
A lot of my email is a waste of time.
Emails are distratcing.
I like doing it "my way"

Assistant

I can't train you explicitly
I want you to learn my priorities
You need to do things my way.

Employees

We want to
retain autonomy

ϟ We do things
without telling you.

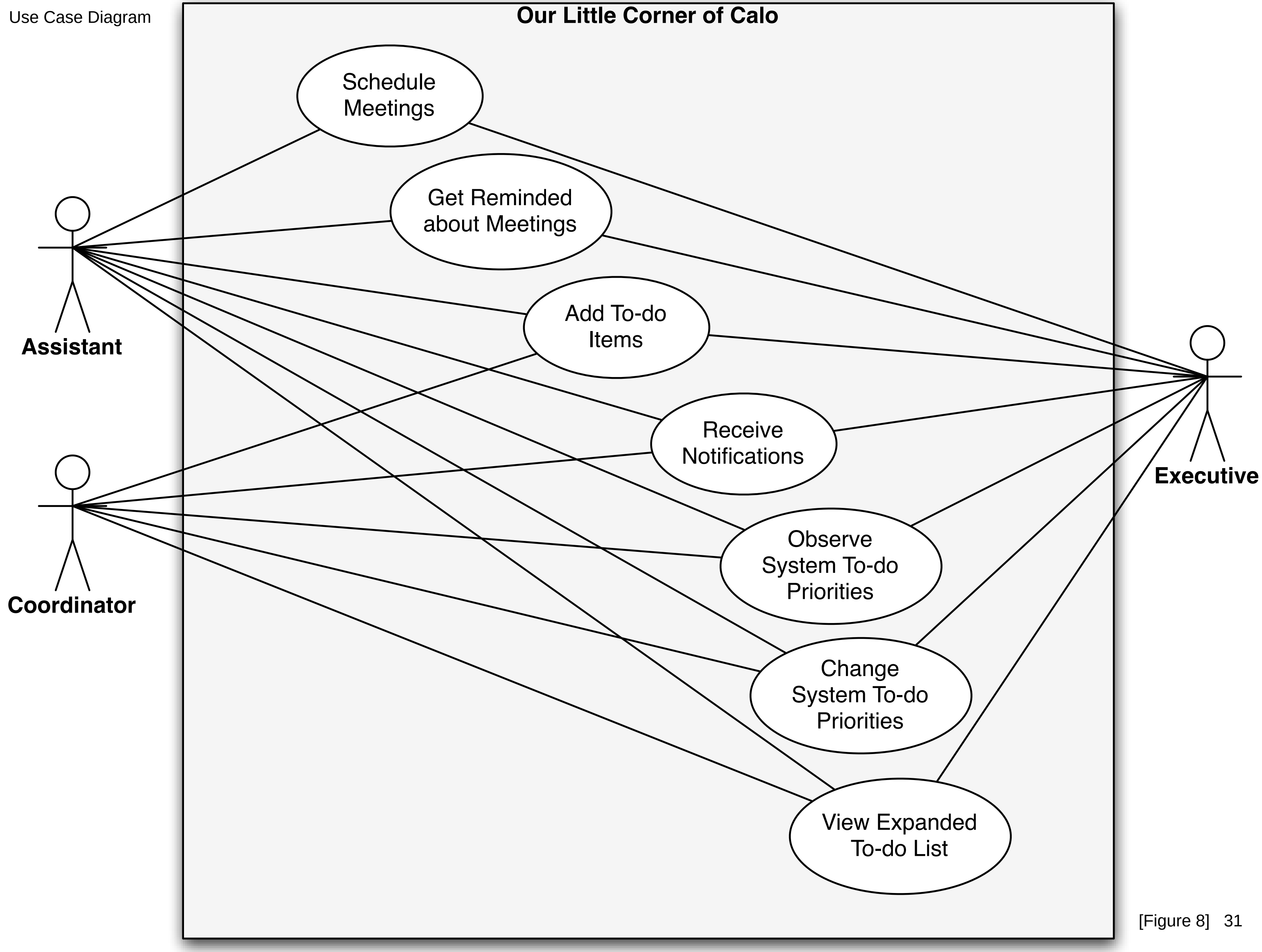
Stakeholders
(customers, organization, etc.)

You must conform
to our preferences

ϟ Our preferences are
not your preferences

[Figure 7]

Use Case Diagram


Our Little Corner of Calo
Schedule Meetings
Get Reminded about Meetings
Add To-do Items
Receive Notifications
Observe System To-do Priorities
Change System To-do Priorities
View Expanded To-do List
Assistant
Coordinator
Executive

[Figure 8]

7 Bibliography

1. Adamczyk, P.D., Bailey, B.P., A Method and System for Intelligent Interruption Management. in Task Models and Diagrams, (2005).
2. Avrahami, D., Hudson, S. E., QnA: Augmenting an Instant Messaging Client to Balance User Responsiveness and Performance. in ACM Conference on Computer Supported Cooperative Work, (2004).
3. Bao, X., Herlocker, J., Dietterich, T., Fewer Clicks and Less Frustration: Reducing the Cost of Reaching the Right Folder. in IUI, (Sydney Australia, 2006).
4. Barthelmess, P., Kaiser, E., Lunsford, R., McGee, D., Cohen, P., Oviatt, S Human-Centered Collaborative Interaction.
5. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool.
6. Berry, P., Myers, K., Uribe, T., and Yorke-Smith, N., Task Management under Change and Uncertainty: Constraint Solving Experience with the CALO Project. in CP 2005 Workshop on Constraint Solving under Change and Uncertainty, (Sitges, Spain, 2005).
7. Berry, P., Peintner, B,. Yorke-Smith, N. , Bringing the User back into Scheduling: Two Case Studies of Interaction with Intelligent Scheduling Assistants. in AAAI Spring Symposium on Interaction Challenges for Artificial Assistants, (Standford, CA, 2007).
8. Berry, P.M., Gervasio, M., Uribe, T, Pollack, M., and Moffitt, M., A Personalized Time Management Assistant. in AAAI Spring Symposium Series, (Stanford, CA, 2005).
9. Berry, P.M., Gervasio, M., Uribe, T., Myers, K., and Nitz, K., A Personalized Calendar Assistant. in AAAI Spring Symposium Series, (Stanford, CA, 2004).
10. Beyer, H., Holtzblatt, K. Contextual Design: Defining Customer-Centered Systems. Morgan Kauffmann, San Francisco, 1998.
11. Biehl, J.T., Bailey, B. P., Improving Interfaces for Managing Applications in Multiple-Device Environments. in AVI, (2006), 35-42.
12. Boardman, R., Sasse, M. Stuff Goes into the Computer and Doesn't Come Out. in A Cross-tool Study of Personal Information Management, 2004.
13. Cheyer, A., Park, Jack, Giuli, Richard., IRIS: Integrate. Relate. Infer. Share. in ICSC (Galway, Ireland, 2005).
14. Chklovski, T., Gil, Y., Enhancing Interaction with To-Do Lists Using Artificial Assistants. in AAAI Spring Symposium on Interaction Challenges for Artificial Assistants, (Stanford, CA, 2006).
15. Conley, K., Carpenter, J., Towel: Towards an Intelligent To-Do List. in AAAI, (2006).

16. Dabbish, L.L., Kraut, R., Fussell, S., Kiesler, S., Understanding Email Use: Predicting Action on a Message. in ACM Conference on Human Factors in Computing Systems, (Portland, Oregon, 2005), 691-700.
17. Dietterich, T., Kaelbling, L., Marx, Z., Rosenstein, M. , Transfer Learning with an Ensemble of Background Tasks. in NIPS Workshop on Transfer Learning, (Whistler, BC, 2005).
18. Ehlen, P., Niekrasz, J., Purver, M., A Meeting Browser that Learns. in AAAI, (2007).
19. Faulring, A., Myers, B. A., Enabling Rich Human-Agent Interaction for a Calendar Scheduling Agent. in Conference on Human Factors in Computing Systems Extended Abstracts, (Portland, Oregon, 2005).
20. Fogarty, J., Ko, A. J., Aung, H. H., Golden, E., Tang, K. P., Hudson, S., Examining Task Engagement in Sensor-Based Models of Human Interruptability. in ACM Conference on Human Factors in Computing Systems, (Portland, Oregon, 2005), 331-340.
21. Garera, N., Rudnicky, A. I., Towards a Personal Briefing Assistant. in AAAI Spring Symposium, (2005).
22. Grosz, B., Sidner C. Plans for Discourse. in Intentions in Communication, Bradford Books/MIT Press, Cambridge, MA, 1990, 417-444.
23. Henderson, S. Genre, Task, Topic, and Time: Facets of Personal Digital Document Management.
24. Huang, X., Oviatt, Sharon L., Toward Adaptive Information Fusion in Multimodal Systems. in MLMI, (2005), 15-27.
25. Iqbal, S.T., Bailey B. P., Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks. in ACM Conference on Human Factors in Computing Systems, (2007).
26. Kaiser, E., SHACER: a Speech and Handwriting Recognizer. in ICMI, (2005).
27. Kaiser, E., Demirdjian, D., Gruenstein, A., Li, X., Niekrasz, J., Wesson, M. and Kumar, S., Demo: A Multimodal Learning Interface for Sketch, Speak and Point Creation of a Schedule Chart. in ICMI, (State College, PA, 2004), 329-330.
28. Kaiser, E.C., Dynamic New Vocabulary Enrollment through Handwriting and Speech in a Multimodal Scheduling Application: Making Pen-Based Interaction Intelligent and Natural. in AAAI Symposium Technical Report FS-04-06, (Arlington, VA, 2004), 85-91.
29. Kaiser, E.C., Barthelmess, P., Huang, X. Demirdjian, D. A, Demonstration of Distributed Pointing and Referencing for Multimodal Collaboration Over Sketched Diagrams. in ICMI, (Trento, Italy, 2005).
30. Kaye, R., Karam, G., Cooperating Knowledge-Based Assistants for the Office
31. Klimt, B., Yang, Y., The Enron Corpus: A New Dataset for Email Classification Research. in European Conference on Machine Learning, (2004).

32. Kozierok, R., Maes, P., A Learning Interface Agent for Scheduling Meetings. in IUI, (1993).
33. Krause, A., Siewiorek, D. P., Smailagic, A., Farringdon, J., Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing. in IEEE International Symposium on Wearable Computers, (New York, NY, 2003), 88-97.
34. Lerman, K., Gazen, C., Minton, S. and Knoblock, C. A., Populating the Semantic Web. in AAAI 2004 Workshop on Advances in Text Extraction and Mining, (2004).
35. Lunsford, R., Kaiser, E., Barthelmess, P., Huang, X., Managing Extrinsic Costs via Multimodal Natural Interaction Systems. in CHI, (Montréal, Québec, Canada, 2006).
36. Maes, P. Agents that Reduce Work and Information Overload. Communications of the ACM, 37 (7). 30-40.
37. Maheswaran, R., Tambe, M., Varakantham, P., and Myers, K. Adjustable Autonomy Challenges in Personal Assistant Agents: A Position Paper. Autonomy, 2003.
38. Martin, I., Jose, J.M., Fetch: A Personalised Information Retrieval Tool. in RIAO, (2004).
39. Modi, P.J., Veloso, M., Smith, S., Oh, J., CMRADAR: A Personal Assistant Agent for Calendar Management. in Agent Oriented Information Systems, (2004).
40. Papazoglou, M.P. Agent-oriented Technology in Support of e-business. Communications of the ACM, 44 (4). 71-77.
41. Pollack, M., Weber, J., Effective Interaction Strategies for Adaptive Reminding. in AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants, (2007).
42. Rich, C., Sidner, C., COLLAGEN: When Agents Collaborate with People. in International Conference on Autonomous Agents, (1997).
43. Schurr, N., Varakantham, P. Bowring, E., Tambe, M., Grosz. B. Asimovian Multiagents: Applying Laws of Robotics to Teams of Humans and Agents. AAMAS.
44. Shen, J., Li, L., Dietterich, T., Herlocker, J., A Hybrid Learning System for Recognizing User Tasks from Desktop Activities and Email Messages. in International Conference on Intelligent User Interfaces, (Sydney, Australia, 2006), 86-92.
45. Tambe, M., Bowring, E., Pearce, J.P., Varakantham, P., Scerri, P., Pynadath, D., Electric Elves: What Went Wrong and Why. in AAAI Spring Symposium on What Went Wrong and Why, (Stanford, CA, 2006).
46. Tomasic, A., Zimmerman, J., Simmons, I., Linking Messages and Form Requests. . in IUI, (2006).