Augmenting Shared Personal Calendars

Joe Tullio, Jeremy Goecks, Elizabeth D. Mynatt, David H. Nguyen

College of Computing
GVU Center, Georgia Tech,
Atlanta, GA, 30332-0280
{jtullio,jeremy,mynatt,dnguyen}@
cc.gatech.edu

ABSTRACT

In this paper, we describe Augur, a groupware calendar system to support personal calendaring practices, informal workplace communication, and the socio-technical evolution of the calendar system within a workgroup. Successful design and deployment of groupware calendar systems have been shown to depend on several converging, interacting perspectives. We describe calendar-based work practices as viewed from these perspectives, and present the Augur system in support of them. Augur allows users to retain the flexibility of personal calendars by anticipating and compensating for inaccurate calendar entries and idiosyncratic event names. We employ predictive user models of event attendance, intelligent processing of calendar text, and discovery of shared events to drive novel calendar visualizations that facilitate interpersonal communication. In addition, we visualize calendar access to support privacy management and long-term evolution of the calendar system.

Keywords: Groupware calendar system, user modeling, social visualization, text classification, privacy management.

INTRODUCTION

Groupware calendar systems (GCSs) are electronic calendars capable of being shared across a network. They have been in institutional use for decades, and represent one of the earliest groupware applications. Modern GCSs often include access control for privacy management, facilities for meeting scheduling, and integration with other artifacts such as address books or to-do lists.

Despite widespread use in many organizations, GCSs face the same obstacles to successful deployment as many other groupware applications [7]. For organizations new to GCSs, they represent potentially dramatic changes in the way employees maintain and use their calendars.

Recent work by Palen and Grudin [19][20] has investigated GCS use at two large organizations and identified a number of factors that contribute to their success or failure. These factors originate from several different perspectives on GCS use (identified in [19]) as well as the ways in which these perspectives interact. The needs of individual calendaring practices, the use of GCSs as tools for interpersonal communication, and the co-evolution of the calendar technology with the social processes built around it are all important considerations for the design and deployment of groupware calendars.

Given these perspectives, past research and the work presented here point to several work practices requiring support from the GCS. Personal calendaring practices, especially those of mobile users, require a high degree of flexibility in both the manipulation of calendar events and the language used to describe them. To support informal communication, GCSs must let users browse their colleagues' calendars to assess availability. Users must possess knowledge of how their calendars are accessed by others to effectively manage privacy settings.

To directly support these work practices, we have developed the Augur system. Augur is an open model GCS for workgroups that considers the multiple use perspectives critical to successful deployment. We support flexible, personal calendar artifacts by leveraging existing mobile calendars on PDAs and providing for individual strategies that treat the calendar as a surrogate memory, not as a precise schedule of planned attendance. We support interpersonal communications by using predictive modeling and visualizations that facilitate estimates of coworker availability at shared events. Lastly, by logging and visualizing calendar access patterns to their owners, we provide an awareness of group calendar use that allows individual privacy settings to co-evolve with the social environment of the GCS.

In the following section, we explain how the results of past work in groupware calendaring have motivated the design of Augur. We then discuss the system architecture, predictive user models, and text processing techniques employed by Augur. Next, we describe the system's visualizations for facilitating informal communication. We follow with additional visualizations intended to inform privacy management decisions and provide social

UIST'02, October 27-30, 2002, Paris, FRANCE.

accountability to calendar sharing. Finally, we discuss directions for future work in this domain and conclude.

RELATED WORK AND AUGUR DESIGN

Most influential to the Augur design are the studies of calendar use performed by Palen and Grudin at Sun Microsystems and Microsoft [19][20]. These studies were able to identify several facets of successful groupware calendar systems. Some aspects, such as a common infrastructure, managerial support, and peer pressure, focus on organizational properties of the institution. Another, usability, is an important property of the software itself. Other aspects, however, address work practices at the site, and divide design considerations into three interacting perspectives. Below, we describe these perspectives and identify key work practices that can be observed from each.

Single-user calendar

This perspective looks at individual practices of calendaring and scheduling, with personal goals such as temporal orientation, tracking, and reminding considered. Flexibility is necessary, both in the tools that users employ to edit their calendars and in the coding of calendar events. A key motivation for our work is the observation that flexible scheduling practices introduce inaccuracies that hinder colleagues' abilities to make estimates of availability from shared calendars. Treated as a surrogate memory, calendars remind the owner of obligations that must be met even if an appointment will be missed. Also, routine meetings may remain on the calendar despite an overriding, unique event, especially if the calendar interface is cumbersome and invites errors when removing events. An unpublished study at a large computer company found that calendars were often cluttered with recurring appointments that were no longer attended (Erin Bradner, personal communication).

Consequently, the calendar becomes an artifact that may not reflect one's true schedule. Our two-month study of a seven-person workgroup within our academic department shows that only 52% of 381 calendar entries were actually attended. Conflicting appointments also contribute to inaccuracies. 12% of unattended appointments in our study were missed due to attendance at another concurrent event. "All-day" appointments such as holidays often prevent attendance at routine events, and were present on 60 of the 413 user-days in our study (14.5%).

Calendar owners, especially those using mobile PDAs that emphasize individual use in contrast to coordinated corporate systems, are likely to have their own unique ways of representing events. Our system anticipates and compensates for this combination of inaccurate entries and ad-hoc naming to support the interpersonal communication practices described next.

Interpersonal Communication

Here the focus shifts to group-oriented tasks. A common use of shared calendar systems is to find open times for scheduling meetings. While our system permits this use, it emphasizes estimating the availability of co-workers for informal chats, and in particular, finding opportunities for conversation at shared events. These brief conversations are critical components of office work [26].

In this vein, we call attention to the practice we term "ambushing," best illustrated by the collection of brief conversations that occur directly before, after, or sometimes during a shared event such as a meeting or seminar. Our original system prototype [17] was designed specifically in support of this practice. Although the terms are used interchangeably in the literature, we distinguish this practice from "waylaying", where co-workers wait for opportunities to catch an individual at his desk [9][27]. For users who regard their desk time as "quiet time" for productive, heads-down work, waylaying may be viewed as a reason to *not* participate in a GCS [3]. In contrast, users who show their availability at shared events may welcome the opportunity to handle short interactions away from their desk without resorting to email and voicemail channels.

Augur allows inaccurate calendars to remain useful tools for initiating informal communication by employing predictive models of user attendance in conjunction with intelligent text processing. By visualizing the output of these models, we provide an informed view of a user's schedule that enhances a coworker's ability to infer her attendance at upcoming events.

Additionally, we seek to support the management of communication resources at one's disposal. More accurate estimates of availability inform a user's choice of an appropriate communication medium. If it is unlikely that a worker can ambush a colleague on a particular day, he will probably use another method, such as writing email or a scheduling a meeting.

Our emphasis on shared events also leads to support for other informal collaboration practices. For example, estimations of attendance at shared events help users assess the importance of a particular event, either in terms of general interest or in the attendance of specific individuals. Likewise, an awareness of the distribution of colleagues at events throughout the week aids in understanding the actions and priorities of that group.

Socio-technical Evolution

Privacy management is an important practice that often employs a combination of strategies such as access settings and omitted appointments. However, users depend on an awareness of how their personal information is being used by others to determine how they employ these strategies. Our system informs users about accesses to specific calendar entries to aid decisions regarding the sharing of personal information.

Technology is created with a particular social setting in mind. Users develop work practices around the initial design even as new features are added. For example, privacy concerns may increase as a GCS user population expands. Our system's added facilities for managing privacy help to overcome the inertia of a calendar that is open by default.

Users remain aware of their level of privacy even as the system's social environment changes.

SYSTEM DESIGN

The Augur system is comprised of a number of components that process, store, and serve calendar information located in a central relational database (Figure 1). It retrieves user calendar data from PalmOS devices, augments the data with information about attendance likelihood and events co-scheduled by colleagues, and serves that information to web-based visualizations that present the augmented calendar to each user and log calendar accesses.

To harvest calendar information, we have implemented PalmOS conduit software that automatically sends calendar information via FTP to our parsing module upon synchronization with a networked computer. The parsing module reads the PalmOS calendar and updates a table of events in the database whose fields are designed to match those of the VCalendar specification [24]. A user ID number is associated with each event to identify its owner. A second table lists the system's users and their IDs.

Once the latest calendar information is retrieved, our prediction and event matching modules insert additional information into the database. The prediction module uses a Bayesian network to add information about the likelihood of attendance for future events. Each user has a copy of the network that is capable of learning their attendance habits over time. An additional component allows users to provide examples to the system by submitting daily attendance checklists via the web. The event-matching module uses text-processing techniques to identify events from other colleagues' calendars that are likely to represent the same event. These modules are discussed in detail in the following sections.

With current, augmented calendar data now present in the database, web-based visualizations display this information to users. The owner's view displays his scheduled events along with information about whom he might see at those events. Information about accesses to those calendar events is also displayed. A second view, provided for the owner's colleagues, displays his events along with information about the likelihood of his attendance at those events. Additional software logs accesses to the visualizations and stores this information in the database. A detailed description of our visualization designs is discussed later in this paper.

All components are written in the Java language, while the visualizations use a combination of Java Server Pages (JSP), and dynamic HTML (DHTML). Database functionality is provided through the MM implementation of MySQL. Norsys Corp.'s Netica software [18] is used for probabilistic modeling and inference, while the SVMLight support vector machine implementation by Joachims [23] is used to classify calendar events using their text.

The Augur system has been in operation for over two months in a workgroup that includes five students, one research scientist, and one associate professor. Our calendar database now contains over 4600 events. Throughout the day, the workgroup is often dispersed among several campus buildings, and some telecommute a few days per week. Since the group often lacks the physical proximity that promotes informal interactions [14], there is a clear need for the facilities provided by Augur.

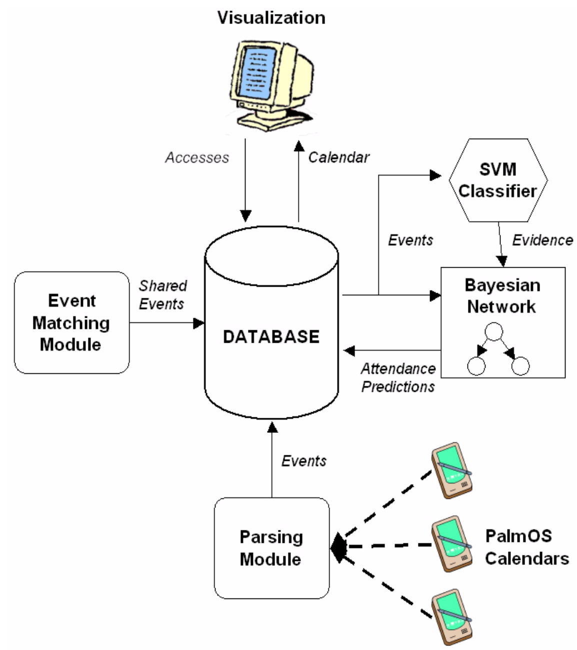

Figure 1: Augur system diagram

PREDICTING EVENT ATTENDANCE

To model the inherent uncertainties in the attendance of users at their scheduled events, our prediction module uses a Bayesian network. Bayesian networks are directed graphs that represent a joint probability distribution over a number of variables that typically exhibit some conditional independence relationships. They provide a compact, descriptive means of encoding uncertainty in systems where we have a fair amount of structure and a store of a priori knowledge about the system in the form of either collected data or experts. They have been used successfully in a number of interactive systems [11] [12] and are useful tools for context-aware applications that must make higher-level inferences under uncertainty from sensed data.

User Model

Figure 2 illustrates our model of the likelihood of a person's attendance at a given event. Nodes in the graph represent variables, and arrows represent direct relationships between two variables. The model specifies the decision to attend as a result of influences from the priority of the event, the priority of a conflicting event, if one exists, and the current availability of the potential attendee. It was created manually from the results of interviews with students and faculty [17].

Event priority depends on whether reminders exist for the event, whether the event recurs, the type of event, and the user's role in the event. "All-day" events such as holidays are also an influence, since they often supersede routine events. Availability is the influenced by the user's location,

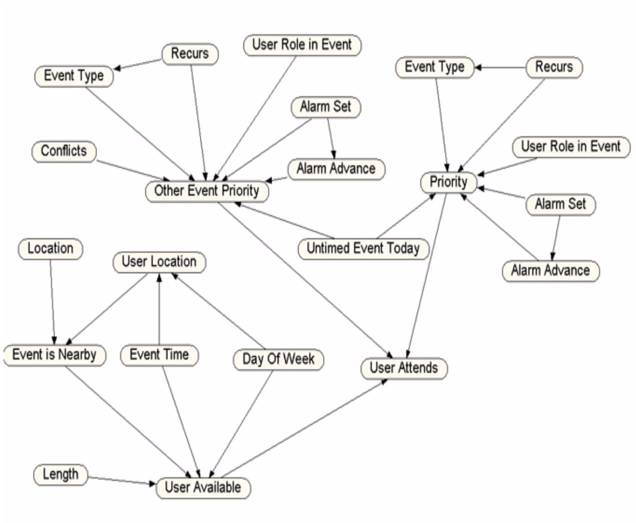

Figure 2: User model of event attendance

the event location, the event time, and the event duration. While we do not yet have the infrastructure in place to provide fine-grained tracking of users across campus and beyond, we have prior information obtained from interviews regarding the location of people throughout the day. While systems using GPS [1] or active badges [6] show promise for schedule prediction, we are hesitant to add it due to cost and privacy constraints.

Since the user can only be in one place at a time, the system must indicate a preference when it encounters conflicting events. Likely priorities of each event are calculated, and the higher-priority event becomes the more likely attended event. While currently only set to handle two conflicting events, the system can be extended to handle more.

In practice, the model contains factors that we are able to sense with varying degrees of confidence. We previously mentioned an optional feedback system that allows users to train the model by filling out an attendance checklist on the web. Since this is an additional burden, however, we have worked to prevent users from resorting to this mechanism where possible. While items such as date and time of day are easily extracted from the calendar event fields, other information such as user location is not yet accessible and must be left uncertain. We instead rely on prior probabilities obtained during the model design phase. Attributes such as event location and event type are not fields in the PalmOS calendar format. To extract this information, we use a novel application of text-processing techniques from the area of intelligent systems.

Support Vectors for Event Classification

The support vector machine (SVM) is a machine learning algorithm that has found great success in the domain of text classification [13]. In simplest terms, SVMs learn a hyperplane classifier that achieves maximal separation between the two classes (true or false). Unseen examples are then tested against this classifier. Although SVMs use a linear algorithm, the optimal hyperplane may not be of a linear form. Therefore, a nonlinear kernel function can be used to map data to a different space where the linear algorithm can be applied.

For text classification, we represent each calendar entry as a feature vector. Each unique word over all event descriptions corresponds to a feature in the vector. Its value is the number of occurrences of that word in the current event description, scaled by its inverse document frequency (IDF). The IDF of a word *w* is defined as:

$$IDF(w) = \log\frac{n}{DF(w)}$$

Where $DF(w)$ is the number of event descriptions in which the word appears and n is the number of documents.

We have trained several SVM models to classify calendar events by their location and type. Event locations have four possibilities: one of three campus buildings (CCB, CRB, or ResLab) or other. Event types include courses, seminars, individual meetings, group meetings, office hours, and other. We conducted an experiment using 1000 labeled calendar events, where 700 were used for training and 300 for testing. By training SVM models using polynomial kernels of degrees one, two, three, four, and five, we created binary classifiers for each possible event location and type. The binary classifiers for location achieved accuracies ranging from 85% for the CCB location to 98% for ResLab. Those for event type had accuracies ranging from 89% for Other to 99% for Office Hours. When combined together, we were able to correctly classify event type 80% of the time, and location 82% of the time.

We find these results encouraging and capable of substantial improvement. Many of the training examples were somewhat dated calendar entries corresponding to events that happened several years ago. In some cases, the events were created before the user even arrived at Georgia Tech and used an entirely different jargon for describing event attributes. We feel that as the database accumulates more recent events, we will be able to institute a cutoff age for training examples and thus have more relevant training data.

IDENTIFING CO-SCHEDULED EVENTS

A critical feature of Augur is the ability to show a user that colleagues are also planning to attend the events he has scheduled. To provide this information, our system must identify *co-scheduled* events, which are events that multiple users have scheduled on their calendars. At the lowest level, the system must identify calendar entries across users' calendars that represent the same event.

However, individuals enter events into their calendars using personal and often idiosyncratic coding styles. Thus, the same event is often represented in many different ways across a set of personal calendars, and it is difficult for a system to automatically determine if any two entries represent the same event. For instance, each week there is a GVU Brown Bag lecture on our campus, and the individuals using our system have entered this event on their calendars using the following descriptions: 'GVU Brown Bag', 'GVU brownbag seminar', 'brown bag', and 'gvu bb.'

We have extended an existing text processing algorithm to identify calendar entries in our system that represent co-scheduled events. We first discuss the algorithm we use and then describe our extensions to identify entries that represent co-scheduled events.

TF-IDF Algorithm

Augur's entry-matching module (the EMM) uses the 'Term Frequency - Inverse Document Frequency' (TF-IDF) text processing algorithm [22] as its basis. As in SVMs, documents are represented as normalized, weighted 'document vectors'. The TF-IDF algorithm uses a set of predetermined keywords. If there are N keywords, then each document vector is N-dimensional. Each vector element is the weight that corresponding keyword carries in describing the document.

For a particular document vector, each element's weight is the TF-IDF value for that element's corresponding keyword with respect to the document. A keyword's term frequency (TF) is the number of times it appears in the document; we previously discussed how to compute a term's IDF value. The value of a document vector's i-th element is the result of multiplying the i-th keyword's TF for the document with the term's IDF:

A keyword's influence in describing a document is directly proportional to the term's TF and inversely proportional to the term's IDF. The dot product of two document vectors indicates how similar the documents are; a larger dot product corresponds to greater similarity.

Event Matching

Augur's EMM discovers co-scheduled events by matching textually similar and temporally synchronized calendar entries across calendars in the system. The EMM uses all entry descriptions stored in the system as the document collection for the TF-IDF algorithm. The module creates a list of keywords by parsing each event description and adding all the words in the description to the keyword list. After determining the keyword list, the module computes an IDF value for each keyword.

The EMM then creates a 'similarity threshold' based on the average similarity between past temporally synchronized calendar entries; synchronized entries start and end at the same time. The EMM uses the average similarity of past events as a baseline and sets the similarity threshold to 40% of this baseline. Finally, the EMM determines synchronized entries whose similarity exceeds the threshold to represent a co-scheduled event.

Design challenges

We encountered two principal challenges when designing the EMM: (a) creating a keyword list and (b) setting the similarity threshold.

We chose the keywords to be all the words that appeared in all calendar entries; there are numerous advantages to such a decision. Automatically creating the keyword list provides maximum flexibility for the system, and a list of predefined keywords would be difficult and tedious for an administrator to maintain. It is also unclear how an administrator would choose a relevant list of keywords. The module currently uses 3191 keywords.

Determining the similarity threshold was challenging because it is not clear what the threshold should be. The similarity of past synchronized entries provides a baseline from which to start, but setting the threshold to this value leads to inaccurate results. The similarity of temporally correlated entries follows a largely bimodal curve; most entries are not similar at all because they represent different events, but some entries are very similar because they represent the same event. Entries whose similarity falls near the average similarity, however, often represent the same event. Often the similarity between the two entries is the result of each entry including a person's name or the event's location.

Hence, it is necessary to set the threshold below the average similarity in order to correctly identify co-scheduled events whose entries that fall at or just below the average similarity. We set the similarity threshold to be only 40% of the average threshold because this value yields the most accurate matches for our data. We would like to explore methods to dynamically change the threshold as the EMM searches for co-scheduled events. We are particularly interested in using recent matches of co-scheduled events to inform future matches.

We recorded statistics on the EMM's accuracy, its false positive rate, and its false negative rate over the course of a month. Augur contained data for 7 users during the month studied; 491 entries were scheduled for the month, and our system matched 146 entries. There were 17 entries scheduled and 4 matches on average per day. The module correctly identified approximately 94% of all correct matches during the month. The module incorrectly labeled 4% of matching entries; the module's false positive rate was 14%, and its false negative rate was 6%.

The EMM more accurately matches recurring events than one-time events. We hypothesize that the workgroup using Augur shares a common language that they use to discuss recurring events and are more likely to describe such events similarly in their calendars than one-time events. Recall the example that we discussed earlier, the GVU Brown Bag lecture. Even though users identified the event differently, almost all users include the word 'GVU' or the word 'brown' in their entries. In contrast, three users described a recent guest lecture on our campus as 'special lecture', 'Donald Prosnitz', and 'Science, Technology, and the Justice System'.

Augur is designed to support a workgroup rather than a complete organization comprised of numerous workgroups. The calendar data we used in this experiment is taken from an existing workgroup, and we believe the data is representative of the data we would find in other workgroups. We hypothesize that most workgroups share a common language, and the EMM utilizes a shared language to find co-scheduled events. We plan to explore this hypothesis in future work.

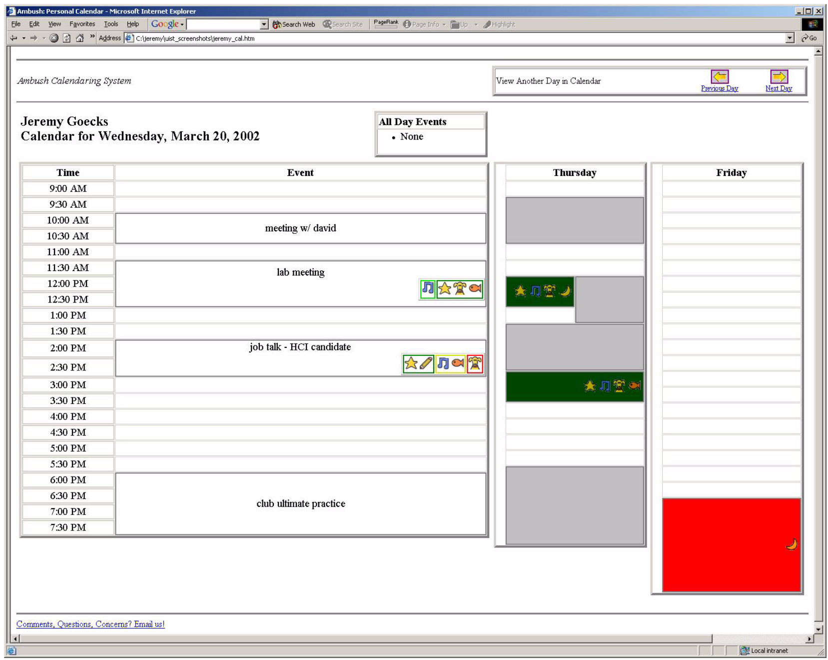

Figure 3: The augmented daily calendar

One-on-one meetings are an interesting example of co-scheduling; the EMM cannot match entries representing a one-on-one meeting using the TF-IDF algorithm because the entries representing the meeting are quite different. One entry might be 'Meet w/ Jeremy,' and the other entry may be 'Meet w/ Beth' or simply 'Beth.' We match one-on-one meetings by finding two temporally synchronized entries that both include the name of the colleague who made the other entry. The EMM identified 100% of one-on-one meetings in our data set using this method.

Finally, the EMM's high false positive rate (14%) is troubling, but this result is likely caused by the small size of our data set. Most false positives occur because a particular, relatively rare word present in both entries boosts the similarity of the entries above the threshold; recall that rare words boost similarity faster than common words in the TF-IDF algorithm. For example, "dinner" is relatively rare word in our current data set, and the EMM has incorrectly matched dinner entries in our data set. Increasing the size of our data set will help to alleviate this problem by reducing the weight of such words.

SUPPORTING INTERPERSONAL COMMUNICATION

We previously described the need for GCSs to support the following interpersonal communicative practices: ambushing, resource management, assessing meeting importance, and workgroup awareness. However, most GCSs only support such practices to a limited extent. In this section, we describe our calendar visualizations and discuss how they support these practices. Finally, we discuss the challenges we encountered in designing these visualizations, as well as their scalability.

Beard et al [2] developed a novel tool for scheduling meetings that mapped manually-entered event priorities to transparency. Mackinlay et al [16] presented a 3D calendar visualizer that displayed free/busy times across a workgroup. Our work builds on these ideas by supporting additional social practices and using an infrastructure of predictive user models.

The Augmented Daily Calendar

Most calendar systems have the notion of a daily calendar view; this view presents the user's scheduled events for a day in an hour-by-hour, block format. We have augmented this view with additional information that supports the four interpersonal communication practices discussed above. The augmented daily calendar is the principal Augur interface; Figure 3 shows a screenshot of the calendar.

We augmented an individual's traditional daily calendar in numerous, subtle ways to incorporate information about which colleagues have scheduled the same events. The events on a worker's calendar are augmented with a list of icons that indicate which of the worker's colleagues have co-scheduled the event. Each icon represents a particular colleague, and a colleague's icon is displayed within an event on the calendar if the colleague has co-scheduled that event. In the calendar shown in Figure 3, the user can see that four other colleagues have co-scheduled the 'lab meeting' event on their calendars.

We combine three techniques to show the likelihood that a colleague will attend a scheduled event. First, icons are arranged horizontally left-to-right within an event according to the colleague's attendance likelihood for the event. Second, we map an icon's opacity to the attendance likelihood of the colleague; the more opaque a colleague's icon is, the more likely the colleague is to attend the event. Finally, we cluster colleagues in an event based on their attendance likelihood using colored boxes; the color of the

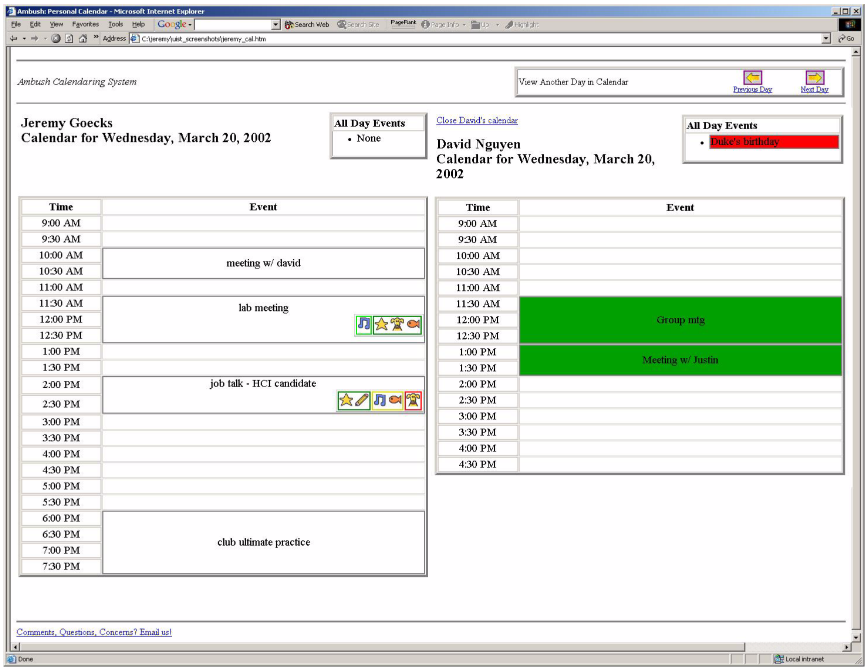

Figure 4: Viewing a colleague's calendar

box around an icon group indicates the attendance likelihood of the colleagues in that box. For example, a bright green box surrounds colleagues' icons that are very likely to attend the event.

The color groups are bright green, green, yellow, red, bright red in descending order of attendance likelihood. We use three techniques to represent attendance probabilities because each technique affords a different understanding of the data; we discuss these affordances shortly.

The Bar Calendar

To the right of the daily calendar are visualizations of the worker's calendar for the upcoming two days, which we call 'bar calendars'. The goal of these visualizations is to provide awareness of the user's upcoming schedule, including information present in the augmented calendar. As in a traditional daily calendar, the bar calendar represents events as blocks that span the event's scheduled duration. However, the bar calendar does not display the events' descriptions.

Event blocks in the bar calendars are colored to indicate the overall popularity of an event; we again use a green, yellow, and red color palette to color the bar calendar's event blocks. An event's popularity is sum of the attendance probabilities of all colleagues who have scheduled the event. Hence, events where the worker is likely to see many colleagues are colored green, events where the worker is likely to see a few colleagues are colored yellow, and events where the worker is unlikely to see any colleagues are colored red. Events scheduled only on the worker's calendar are colored light gray.

As in the daily calendar, we place icons in bar calendar event blocks to indicate which colleagues also have scheduled events that are on the user's schedule. We again use left-to-right ordering and icon opacity to indicate the likelihood that a colleague will attend an event. However, we do not use persistent colored boxes in the bar calendar to indicate attendance likelihood because the event's colored background makes it difficult to determine the boxes' colors.

Interactions

Up to now we have discussed the information that the user can obtain just by looking at his calendar. The user can also interact with the calendar to obtain more information about his colleagues' calendars. When the user mouses over an icon on his daily calendar, a menu pops up (Figure 5). This menu identifies the colleague using his name and a small picture, indicates how likely the colleague is to attend the event, and provides a hyperlink to the colleague's calendar.

When the user clicks on the hyperlink, an animation shrinks the user's calendar, hides the user's bar calendars, and displays the colleague's calendar to the right of the user's daily calendar (Figure 4). This allows the user to easily compare schedules and plan communication with the colleague accordingly. The event blocks on the colleague's calendar are colored to indicate the likelihood that the colleague will attend the events; we again employ the green, yellow, red color scheme used throughout the calendar.

The user can also interact with the bar calendars to obtain more information than is immediately visible. When the user mouses over a colored block in a bar calendar, the

Figure 5: Pop-up menu for a colleague

block's background changes to white and the block's border assumes the block's color. In addition, colored boxes are placed around the event's icons when the user mouses over the block. These boxes are identical in shape, color, and meaning to those in the daily calendar; the icons in the block function identically to the icons in the daily calendar.

Supporting Communication Practices

The augmented calendar supports the interpersonal communication practices previously discussed. We consider each practice in turn:

Ambushing. Each icon on the calendar indicates a time when both the worker and one of her colleagues may be in the same physical location; this time is an opportunity for her to ambush the colleague. She can also scan her daily calendar and the bar calendars to determine what events present many ambushing opportunities, or opportunities to be ambushed.

Resource management. The information on the augmented calendar helps users determine how best to communicate with their colleagues. If a user needs to speak with a colleague about a matter related to an upcoming deadline, he can view the calendar to determine if an opportunity exists to ambush the colleague before the deadline. If not, he can send email to the colleague or set up a formal meeting to discuss the matter.

Determining Meeting Importance. Grudin and Palen have observed that workers often determine a meeting's importance based on who is planning to attend the meeting [8]. The augmented calendar indicates who is likely and unlikely to attend an event, allowing a user to assess the event's importance and plan attendance accordingly.

Workgroup Awareness. A user can scan the augmented calendar and obtain a general awareness of his workgroup. A significant amount of green on a worker's calendar indicates that he and his colleagues are coordinated in their work.

Design Challenges

We encountered multiple challenges when designing the augmented calendar. We briefly discuss them here.

Facilitating normal calendar use. Workers typically interact with their calendars in brief, frequent instances during the day. We have designed the calendar to support these interactions. The information on the augmented calendar is perceptually salient and easily understood; the calendar's interactions are contextually-driven and lightweight.

Representing colleagues. We use non-photorealistic cartoon icons to represent colleagues in the calendar because they provide a number of desirable qualities. The icons are simple, require little screen real estate, and do not demand the user's visual attention, yet they are perceptually salient when focused on. The mappings between colleagues and the icons that represent them are not difficult to learn if the number of colleagues represented on a worker's calendar is small. We discuss why we believe this to be the case below.

Representing attendance likelihoods. We represent attendance likelihoods using three different, overlapping techniques because it is critical data. Each representation affords a different understanding of the same data. An icon's opacity indicates the likelihood a colleague will attend the event. The left-to-right ordering of icons, which is based on attendance likelihood, lets the user easily compare likelihoods among colleagues who have co-scheduled an event. Finally, the colored boxes provide the user with an overview of the co-scheduled events on her calendar; she can easily find the co-scheduled events on her calendar and determine if and how many colleagues are likely to attend the events. Instead of showing numerical estimates of likelihood, these techniques map a likelihood estimate into one of five categories (very likely, likely, ..., unlikely.). This mapping helps the user easily determine how likely a colleague is to attend an event.

Choosing colors. We experimented with many color palettes for representing the attendance likelihood categories and determined that a segmented color palette is most useful when users want to quickly identify events of interest. Quick identification complements the typical use of calendars described above. We chose to use a green-yellow-red palette because this palette is widely known to represent 'yes,' 'maybe,' and 'no' in our culture.

We believe that the Augmented Calendar, with a few extensions, scales to organizations where many thousands of people use a GCS. Currently, the calendar can support a 15-person workgroup; a slight reduction in icon size would allow the calendar to support a 25-person workgroup. We believe it is sufficient for the augmented calendar to support communication with a smaller collection of close colleagues and provide only limited support for communicating with colleagues outside her workgroup. By allowing users to specify their closest colleagues, the augmented calendar could readily support most current workgroups. We plan to add this feature, as our user base is in the process of expanding.

By supporting workgroups over entire organizations, we alleviate the problem of learning the mappings between colleagues and icons. The average user will need to learn relatively few icon mappings. Moreover, workers glance at their calendar multiple times throughout the day, and exposure to the calendar's icons during these numerous viewings will help the user learn quickly.

PRIVACY AND SOCIO-TECHNICAL EVOLUTION

As a group shares their individual calendars and uses them in everyday practice, they form social norms. These norms are manifested in the way people use a system like Augur and are revealed in the trails of social history left by the group [10][25] - who looked at whose calendar, when, how often, etc. The better a group understands the technical workings as well as the social norms of the system, the better they can shape that system to their personal practices, needs, values, and sensibilities. Understanding the socio-technical system will allow users to be better informed when requesting new technical features and functions for the system. It will also help users find social improvisations, if a technical function is missing.

Along with the other visualizations of Augur, the display of social history helps users to better understand the socio-technical system. To get at some of this history, we log and visualize the accesses to calendars made by group members. This feedback gives users awareness and accountability of not only the technical system but also the social system.

Much like peeking one's head into a colleague's office, visualizing calendar accesses will give both parties involved the awareness and accountability in the nature of you-know-that-I-know [5]. From the calendar viewer's perspective, accountability brings in social norms for viewing others' calendars. For example, viewing another's calendar every 10 minutes may be frowned upon. From the calendar owner's perspective, knowing who and how often someone has looked at his calendar may change his comfort level for sharing calendar information. If a trusted colleague is looking at his calendar often, the calendar owner might initiate contact to see if his help is needed.

From our initial user group, we can already see changes in how calendar events are coded when people understand that their calendar information is shared. Because of the experimental nature of the current Augur system, not all technical privacy measures are implemented. Users currently have the option to set an event to either public (viewable to all) or private (viewable only by the owner). When calendars were shared, some people chose to change a subset of their events from public to private. Others rephrased the descriptions of their events. For example, a doctor's appointment changes from "Dr. Monroe" to simply "Monroe".

In our visualizations, we have designed the interface to provide information at three different levels - glance, look, and interactive. Rhodes had a similar notion to lessen disruptions; he called it a "ramping interface" [21]. Glancing will give a small amount of information. Stopping and looking will give more information.

The current implementation of the social history visualization is shown below (Figure 6). An access history box is placed in the top left corner of an event. By glancing at it, users would use perceptual cognition to see the shading of the box. A light grey box outline means very few accesses; the redder the box outline, the more accesses to that event. Using it as a look interface, users could read the number to see the exact number of accesses to that event so far. For the example shown, 15 accesses have been made to this event. We are currently implementing a component that will launch when users click on the access history box. This interactive visualization will give users information such as who has accessed this event and this calendar, when, how often, how recent, from where, and other patterns that arise out of the social history trails.

FUTURE WORK/CONCLUSION

Having performed a pilot deployment of the Augur system, we now look forward to conducting a formal study of the use of Augur and its impact on personal and group calendaring practices. We are currently adding more users within our workgroup to the system, and plan to use shadowing and interviews in our evaluation.

Figure 6: Augmented calendar entry with access count.

The system currently makes few distinctions between the different activity patterns exhibited by users without the aid of learning. For example, professors may have markedly different patterns of activity from students. We are in the process of defining multiple versions of our Bayesian network to account for these differences. We are also developing SVM models that can classify calendar entries by the user's role in those events given their particular activity pattern.

Work at Sun Microsystems [3] has generated a wealth of data on the day-to-day availability and location of office workers. Visualizations of this data facilitate, among other things, comparisons of scheduled time with how that time is actually spent. We are interested in how this data can be used to inform models such as that contained in the Augur prototype to provide abstracted inferences about availability to applications.

We would like to extend the augmented calendar in two ways. Currently, users can easily find a colleague's schedule only if the colleague appears on the user's calendar; we are adding features that let the user readily find the calendar of any colleague. Second, we are experimenting with treemaps to provide longer-term overviews of a user's augmented calendar.

The inherent structure of a Bayesian network can be exploited by an explanatory interface component [15] that illustrates the factors contributing to the model's predictions. This explanation promotes trust in the system, and thus encourages adoption and long-term use. We hope to develop interactive visualizations that allow users to quickly perform initial training of the model.

We have presented Augur, a group calendaring system that retains the flexibility of personal calendaring practices while supporting interpersonal communication via predictive models, intelligent text processing, and visualizations. The system supports co-evolution of the calendar system with its social environment by using visualizations of event accesses to inform privacy management.

ACKNOWLEGMENTS

This research is funded by NSF CAREER award #0092971. We would also like to thank Jonathan Grudin, Erin Bradner, Justin Fuchs, and the members of the Everyday Computing Lab at Georgia Tech.

REFERENCES

1. Ashbrook, D. and Starner, T. (2002) "Enabling Ad-hoc Collaboration Through Schedule Learning and Prediction" In *CHI2002 Workshop on Mobile Ad-hoc Collaboration*.
2. Beard, D. Palaniappan, M. Humm, A., Banks, D. Nair, A. and Shan, Y-P. (1990) "A Visual Calendar for Scheduling Group Meetings" In *Proc. CSCW'90 Conference on Computer-Supported Cooperative Work*, pp. 279-290.
3. Begole, J.B., Tang, J., Smith, R., and Yankelovich, N. "Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups" In *Proc. CSCW'02 Conference on Computer-Supported Cooperative Work.*
4. Bradner, E., Kellogg, W., and Erickson, T. (1999) "The Adoption and Use of 'Babble': A Field Study of Chat in the Workplace" In *Proc. ECSCW'99 European Conference on Computer-Supported Cooperative Work*.
5. Erickson, T. & Kellogg, W. (2000). "Social translucence: An approach to designing systems that support social processes" In *ACM Transactions on Computer-Human Interaction*, 7(1), pp. 59-83.
6. Ford, D.A., Ruvolo, J., Edlund, S., Myllymaki, J., Kaufman, J., Jackson J., and Gerlach, M. (2001) "Tempus Fugit: A System for Making Semantic Connections", In *Proc. Tenth International Conference on Information and Knowledge Management (CIKM)*.
7. Grudin, J. (1994) "Groupware and Social Dynamics: Eight Challenges for Developers" In *Communications of the ACM*, 37(1), pp. 92-105.
8. Grudin, J. and Palen, L. (1997) "Emerging Groupware Successes in Major Corporations: Studies of Adoption and Adaptation", In *Proc. WWCA'97 Int. Conf. On Worldwide Computing and Applications*, pp. 142-153.
9. Gutwin, C., Greenberg, S., Roseman, R. (1996) "Peepholes: Low Cost Awareness of One's Community", In *CHI'96 Companion Proceedings*, pp. 206-207.
10. Hill, W.C. and Hollan, J.D. (1993), "History-enriched digital objects", *Third ACM Conference on Computers, Freedom and Privacy*, pp. 917-20.
11. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998) "The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users" In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence*.
12. Horvitz, E. and Paek, T. (1999), "A Computational Architecture for Conversation" In *Proc. Seventh International Conference on User Modeling*, pp. 201-210.
13. Joachims, T. (1998) "Text Categorization with Support Vector Machines: Learning with Many Relevant Features" In *Proc. European Conference in Machine Learning (ECML)*.
14. Kraut, R., Egido, C. and Galegher, J. (1990) "Patterns of communication in scientific research collaboration" In J. Galegher, R. Kraut, and C. Egido, Eds. *Intellectual Teamwork*, Hillsdale, N.J.: Lawrence Erlbaum Press.
15. Madigan, D., Mosurski, K., and Almond, R. (1996) "Graphical Explanation in Belief Networks" In *Journal of Computational and Graphical Statistics*, 6(2), pp. 160-181.
16. Mackinlay, J., Robertson, G., and DeLine, R. (1994) "Developing Calendar Visualizers for the Information Visualizer". In *Proc. UIST'94 Symposium on User Interface Software and Technology*, pp. 109-118.
17. Mynatt, E. and Tullio, J. (2001) "Inferring Calendar Event Attendance" In *Proc. IUI2001 Conference on Intelligent User Interfaces*, pp. 121-128.
18. Norsys Corp, http://www.norsys.com/
19. Palen, L. (1999) "Social, Individual & Technological Issues for Groupware Calendar Systems", In *Proc. CHI'99*, pp. 17-24.
20. Palen, L. and Grudin, J. "Discretionary Adoption of Group Support Software: Lessons from Calendar Applications". To appear in B.E. Munkvold, Ed. *Organizational implementation of collaboration technology*.
21. Rhodes, B. J. (2000) "Margin Notes: Building a Contextually Aware Associative Memory", In *Proc. IUI'00 Conference on Intelligent User Interfaces*, pp. 219-224.
22. Salton, G. (1991). *Developments in Automatic Text Retrieval. Science*, Vol 253, pp. 974-97.
23. SVMLight, http://svmlight.joachims.org/
24. VCalendar specification. http://www.imc.org
25. Viégas, F. and Donath, J. (2002) "PostHistory: Visualizing Email Networks Over Time" In *Proc. of the International Sunbelt Social Network Conference XXII*.
26. Whittaker, S., Frolich, D., and Daly-Jones, O. (1994) "Informal Workplace Communication: What Is It Like and How Might We Support It?" In *Proc. CHI'94*, pp. 131-137.
27. Whittaker, S. (1995) "Rethinking video as a technology for interpersonal communications: theory and design implications." *International Journal of Man-Machine Studies*, 42, pp 501-529.