Inferring Calendar Event Attendance

Elizabeth Mynatt and Joe Tullio

Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332, USA
[mynatt, jtullio]@cc.gatech.edu; Tel: +1-404-385-1102

ABSTRACT

The digital personal calendar has long been established as an effective tool for supporting workgroup coordination. For the new class of ubiquitous computing applications, however, the calendar can also be seen as a sensor, providing both location and availability information to these applications. In most cases, however, the calendar represents a sequence of events that people could (or should) attend, not their actual daily activities. To assist in the accurate determination of user whereabouts and availability, we present Ambush, a calendar system extension that uses a Bayesian model to predict the likelihood of one's attendance at the events listed on his or her schedule. We also present several techniques for the visual display of these likelihoods in a manner intended to be quickly interpreted by users examining the calendar.

Keywords

Context-aware, Bayesian networks, groupware calendar systems, calendars, informal meeting scheduling, visualizing uncertainty

INTRODUCTION

The digital calendar is a common artifact found on the PDA's, mobile phones, and/or PC's of most office workers. In the workplace, a number of benefits are attained through the ability to browse the calendars of others. To this end, a great many workplaces have seen the introduction and adoption of groupware calendar systems (GCS's). These systems make it possible for calendar information to be made publicly available in some form to members of the same workgroup or institution. Such systems commonly facilitate such activities as meeting scheduling, temporal orientation, and reminding. Recently, a study by Palen [20], while confirming the popularity of the aforementioned calendar tasks, identified a number of additional uses for GCS's. These uses included location detection, record tracking, and peer judgment, among many others. Office workers will often use public calendar information to locate a colleague. In addition, users may track their own schedules to identify trends in their activities. People will often browse the calendars of coworkers to assess the quality of time corresponding to free/busy blocks in the schedule. They may assess a coworker's workload, attempt to determine activities undertaken between scheduled events, or estimate the priorities of those events to the coworker. Therefore, an accurate representation of the events to be attended improves a coworker's ability to infer this information.

IUI'01, January 14-17, 2001, Santa Fe, New Mexico, USA.

Informal Communication

The location information available through public calendars is invaluable to coworkers who are attempting to 'drop in'[1] on a colleague, whether through office visits or encounters at some other event. In our own academic department, as well as in other research settings [6] [7], a great deal of work is accomplished through this type of informal communication [14]. While meetings and courses make it difficult to catch up with students and faculty, deviations from the schedule outlined on a public calendar exacerbate the problem. To this end, we are pursuing the development of applications that provide a more accurate picture of coworker location and availability. We make use of the calendar, in addition to other contextual information, as a means of facilitating informal office visits.

In the research community, public calendar information is being incorporated into a number of wearable and ubiquitous computing applications that treat the calendar as an additional piece of context. Yan's context-aware office assistant [23] uses a person's calendar to inform a personified agent of available meeting times for visitors at that person's office door. Applications based on the CLUES message filter [17] use calendar information and other sources in the work environment to determine the relevance of incoming messages. The Bayesian user models developed for the Attentional Systems project at Microsoft Research [10] use calendar information as a means to determine the user's state of attention.

Calendars as Sensors

In all of these cases, we can view the personal calendar not only as an information storage artifact, but a *sensor* that can inform software applications as to the location, availability, and workload of a person. Like many sensors, the calendar is a portable object that can serve multiple components in any number of environments. In addition, the information contained within the calendar is dynamic, requiring periodic updates by programs using it.

1. In our work environment, we good-naturedly refer to this practice as an "ambush". For example, planning to talk to someone at a mutually attended seminar. It is seen as an effective and positive practice.

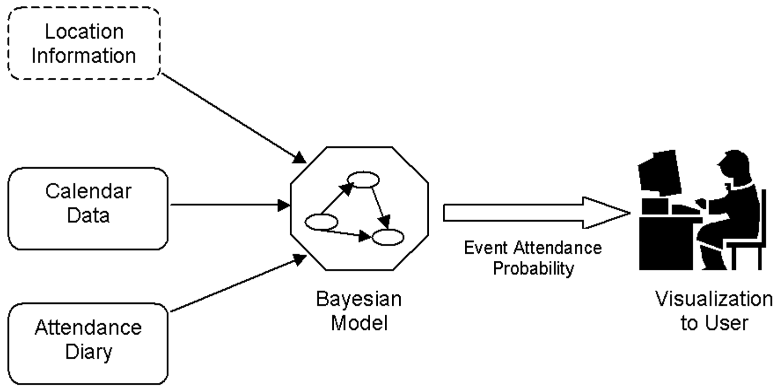

Figure 1: Ambush system diagram

Perhaps the most important similarity between calendars and sensors, however, is the potential for inaccuracies in the information presented to applications. Systems that attempt to use calendar information to locate individuals or determine their availability ignore the actual attendance habits of calendar owners. Two events may have a time conflict, with an important isolated event overriding a routine recurring event. The user may lose interest in a recurring event and neglect to remove it from the calendar. In each case, the calendar will provide incorrect input.

A Probabilistic Calendar

Though the task of predicting which events on a user's calendar will be attended seems infeasible, we can make use of attendance history and attributes of calendar events to present a *likelihood* of attendance to applications. Through informal interviews and a cluster analysis of data elicited from those interviews, we have constructed a Bayesian model of user attendance preferences. Using this model, we have developed Ambush, a system that can predict the probability of a user's attendance at a given future event on his/her calendar.

Figure 1 provides an overview of the system. An "attendance diary" kept by the calendar's owner lets the system use a form of reinforcement learning to improve its predictions over time. Autonomous components scan for new diary entries and use them as evidence to teach the model. Application programmers can then use the model as a sensor to make more informed assumptions about a person's schedule.

We have already noted that our primary motivation is the support of informal visits in the workplace. Given that people have so many uses for public calendars, we have focused our efforts on the design of a simple calendar display program. To this end, we have created several simple visualizations of the uncertainty expressed in the Bayesian model in hopes of effectively conveying the likelihood of a person's attendance at a given event.

At times, the model must make uninformed predictions due to a short learning period or a novel situation. In these cases, the user should be able to access more information about the model for the purpose of diagnosing or verifying its predictions. Therefore, we also attempt to visualize the model's internal influences for users who may be skeptical of its output. The design is loosely based on Kohonen's feature map [12] and visually captures the relationships and degrees of influence of variables within the model. We believe that a major issue for future HCI research in wearable and ubiquitous computing will be the presentation of this type of explanatory information for intelligent applications.

The Ambush system is currently being used at the Georgia Tech College of Computing to provide awareness of faculty whereabouts. This awareness supports the initiation of informal meetings by students, administrators, or other faculty.

A word on privacy, or "Why is he skipping my meeting?"

We understand that the information provided by our system has the potential to be used in a manner that could make some office workers uncomfortable. After all, hurt feelings or misunderstandings might arise when an event's organizer sees that the probability of a particular coworker's attendance is low. We note that our system, since it learns over time, reflects trends in a person's attendance habits that are likely to be noticed by organizers or other attendees.

However, the motivation for this system originated from the calendar owner's perspective. The owner needs to connect with numerous people seeking her input on a daily basis. Any attempt at scheduling all of these brief discussions would be infeasible. Therefore, any support for assisting these informal meetings is welcome.

The prototypes presented later in this paper display information explicitly, and, as with all personal information, must be used only with user consent. Applications that use the system as a source of contextual information must ensure that the user has the option of not making their predicted attendance available, or provide a more qualitative representation.

The next section presents some related work in the areas of informal scheduling, group awareness systems, and learning systems for calendar applications. We then discuss our implementation, including the user model, system architecture, and visualization prototypes of the system's

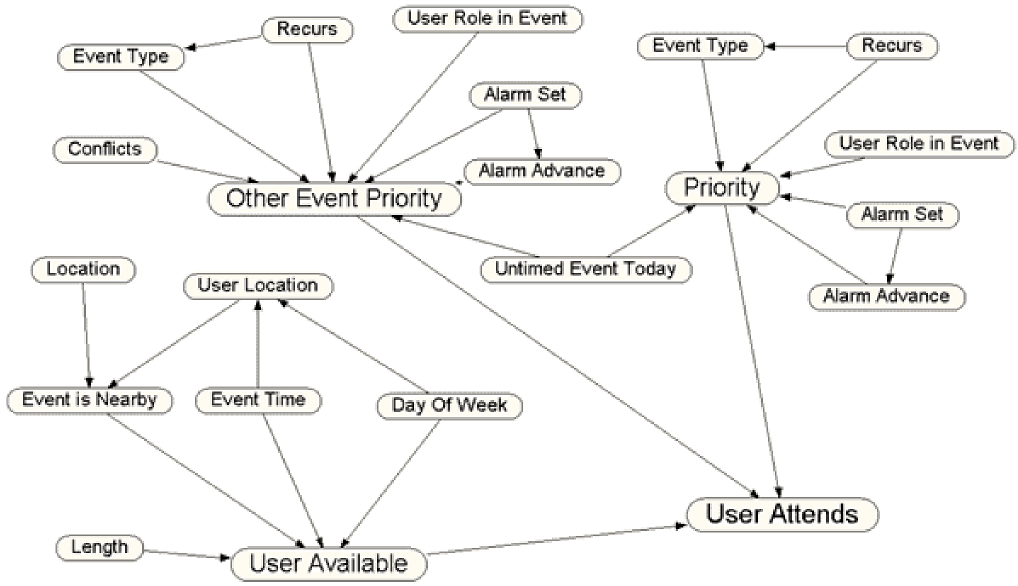

Figure 2: A Bayesian model of event attendance

output for use in calendar browsing. Lastly, we outline future work for the project and conclude.

RELATED WORK

Perhaps the most closely related work is Yan and Selker's Context-Aware Office Assistant [23], a system to manage appointment scheduling at the office threshold. This system used a small interactive application inside the office to obtain the owner's willingness to meet with visitors. Our probabilistic calendar provides a better picture of the owner's availability in the cases where the user does not have feedback from the office owner. These cases might include the owner's absence from the office or passive browsing of his/her calendar from a remote location. Such a calendar would also provide an estimation of the owner's availability to the agent at the door, obviating the need to interrupt those inside the office.

Beard et al [1] describe a visual calendar intended to facilitate the scheduling of group meetings. Their implementation assigns each calendar entry a transparency level corresponding to the user-defined priority of the event, where lower-priority events are more transparent and the highest-priority events are opaque. Meeting scheduling then becomes a matter of overlaying user schedules and finding the least opaque area that could accommodate the meeting. We think that our calendar system could automate the assignment of priority levels without requiring the user to prioritize each entry by hand. In addition, changing notions of a recurring event's priority are reflected in the attendance habits of the calendar's owner, and would subsequently be represented in our model.

A number of systems have been developed to incorporate calendar use information to teach an agent the scheduling habits of its users [18] [13]. It should be noted that while most of the aforementioned systems incorporate some form of learning to facilitate scheduling activities, they do not consider or attempt to predict the attendance of the user at the events scheduled. Therefore, these systems still treat calendars as the actual schedules their users will follow. In addition, our probabilistic calendar is intended to support a broad range of applications, of which meeting scheduling is but one.

Of critical importance to our work is the issue of *trust* in agent-based systems, also addressed by Kozeriok and Maes [13]. By letting an agent learn, either through reinforcement or observation, its inferences become more accurate over time. The user comes to trust its decisions and gradually offload more mundane duties to the agent. The issue of trust is further discussed in a position statement by Kurlander [2], which states that an agent-based system should present a clear model of the input to the user in order to let the user understand the basis for its decisions. Since we have already stated that the task of precise attendance determination is infeasible, we must present a "paper trail" of the agent's reasoning to the user. This explanatory information forms the basis for a novel visualization prototype discussed later in this paper. An important body of work on explanatory techniques for probabilistic inference exists in both verbal [4] [22] and graphical [16] forms.

Research in the field of groupware calendar systems is quite substantial. We have already touched on the excellent study of issues for GCS's by Palen [20]. Other studies that have addressed the social uses of GCS's include work by Mosier and Tammaro [19], and by Ehrlich [5]. These studies, along with our own observations in the workplace, demonstrate a clear purpose for a system that presents users with a more accurate schedule for a coworker.

IMPLEMENTATION

Bayesian Models

To model the inherent uncertainties in the attendance of users at their scheduled events, we developed a Bayesian network to model attendance habits. Bayesian networks provide a compact, descriptive means of encoding uncertainty in systems where we have a fair amount of structure and a store of prior knowledge about the system in the form of either collected data or experts. They have been used successfully in a number of interactive systems [3] [9] [11], and are useful tools for context-aware applications that must make higher-level inferences under uncertainty from sensed data. As we shall see, the inherent structure of a Bayesian model can be incorporated into an explanatory interface component that illustrates the factors contributing to the model's predictions.

User model

Figure 2 illustrates the model used to determine the likelihood of a person's attendance at a given event. We created this model by hand using Norsys Corp.'s Netica belief network software. While certainly not a comprehensive model of all factors that one considers when making the decision to attend, we feel that it captures the major influences and generates reasonable predictions in practice. 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.

To arrive at the network depicted in Figure 2, we conducted informal interviews with both students and faculty within our academic department. This work resulted in a substantial list of items considered by interviewees to have some influence in their decisions to attend an event. With this list in hand, we then worked to define the structure uniting these factors toward a single decision. We decided that the majority of factors listed contributed to an assessment of the event's priority and the person's availability. These were established as meta-criteria in our model, exerting direct influence on the final decision to attend. Other information garnered from interviews concerned the existence of conflicting events. We therefore incorporated an additional branch of the model to handle the existence of such a conflicting event. To determine the variables that would be depicted as nodes in the network, we identified common decision factors that were mentioned by many of the interviewees.

To use the model in practice, we identified those factors that we would be capable of sensing, either now or in the near future, for the purpose of teaching the network through reinforcement. Sensing capabilities were limited by our choice of calendar format, and by our department's current technology infrastructure. Although this step resulted in some paring down of the original network, factors that were determined to be the largest influences (e.g., the person's role in the event, his/her location, all-day events) were kept intact. The remainder of this section provides a more detailed description of the items comprising the network.

The priority of an event is influenced by a number of factors, including the alarm status, recurrence status (the event occurs daily, weekly, yearly, etc.), the type of event (course, seminar, group or individual meeting, etc.), and the user's role in the event (organizer, mandatory attendee, etc.). The presence of "all day", or untimed, events influences the priority of other events on the schedule, since such events typically supersede routine recurring events.

The model specifies availability as the result of influences from the user's location, the event time, and the length of the event. Many students and faculty in our department share their time between three different buildings that are not physically proximal to one another. While we do not yet have the infrastructure in place to provide fine-grained tracking of users across campus and beyond, we do have the ability to determine the building a person is currently in. This coarse location history, combined with prior information elicited through interviews with the user, allow us to obtain probabilities that serve as a reasonable estimate for both location and availability.

The existence of a conflicting event presents an interesting problem. In this case, the user can only be in one place at that time, so a judgment by the system must be made on which event is preferred. Therefore, the system considers the priorities of the current event and the conflicting event, as well as user availability, to determine an appropriate likelihood of attendance for the event. Priorities for both events are determined using the same criteria described earlier. Note that a current limitation of the model is that only one other conflicting event is considered. We plan to extend the model to handle an arbitrary number of conflicting events in the future.

Evaluation

To this point, we have trained the Ambush system on two and a half months worth of calendar data for a single user, or roughly 200 events. Since we had no existing store of data prior to the initiation of the project, the learning time is fairly short. Therefore, we have not yet performed any statistical analysis on the results produced by the system. However, the results of the system have been available to members of our lab and department since the learning period began. Qualitatively, we have noticed trends in attendance habits as probabilities for recurring events have increased or decreased over the course of several weeks. Users are clearly able to observe the relative prospects of attendance for a day's worth of calendar events. The learning algorithm used by our software incorporates a measure of experience, so that probabilities are changed less as more cases are presented to the network. As expected, routine events appear to be stabilizing more rapidly than the more diverse class of special, one-time events. Our future work section discusses some ideas for refining the accuracy of the network and shortening its learning time.

Visualizations

As stated earlier, our initial use for the calendar is to support informal communication in the workplace by giving users an improved perception of the location and availability of a colleague. To this end, we have created several interface prototypes for a graphical schedule viewable by any colleague in the same department as the schedule's owner. The designs incorporate visualizations of the owner's likelihood of attendance at the events listed. These visualizations are intended, at a glance, to convey both the likelihood for a given event as well as the relative likelihoods for an entire day of events.

Prototype 1: Bar Graphs

Our first attempt at a graphical depiction of our calendar's output was to display the likelihood of attendance as a basic bar graph. While simple in design, the prototype clearly shows the relative probabilities of attendance for the day's scheduled events. Additionally, the design is easy to implement in standard HTML, so a simple CGI script can generate the view from the current state of the network and make it available from any graphical web browser.

Prototype 2: Transparency

Figure 3 shows a prototype using a transparency technique similar to that employed in Beard et al's work. Whereas their system associates transparency with a user-assigned priority, our prototype maps transparency to likelihood of attendance. The priority algorithm for their system, which used local minima in the opacity of attendee calendars to identify common periods of availability, could also be adapted to this prototype. A person's most probable choice between two conflicting events can be determined through simple visual inspection. Events with a higher probability of being attended are made more opaque, while an event with a low probability of attendance is still kept legible using the cutoff of 90% transparency given by Harrison et al [8]. Transparency for this prototype has been discretized to eight levels to help users distinguish between relatively close probabilities.

Prototype 3: Feature Map Variation

As with any learning system, the introduction of novel situations will often cause the system to generate inaccurate predictions. In both of the prototypes just presented, the probability of attendance is represented as a single scalar value that may or may not be reasonable to the viewer. Since Bayesian networks provide a more human-readable representation of the system being modeled than "black box" techniques such as neural networks, their underlying structure can be used to provide a better understanding of the process that led to the system's prediction. In addition, the probabilistic relationships between nodes of the network can be used to identify those variables whose evidence will have the most profound effect on another variable's likelihood. Our next prototype attempts to use this information to visualize the most important influences on the network's prediction, allowing the viewer an abstracted view inside the "black box" of the system.

To provide this view, we have designed a variation on Kohonen's feature map [12]. The feature map is the result of an algorithm that maps a set of N-dimensional input objects to a static two-dimensional grid. The two-dimensional nature of the map allows not only a representation of the inputs themselves, but the relationships between inputs as well. The main properties are:

- Inputs that occur more frequently occupy more space at the expense of those occurring less frequently.
- Proximity of inputs on the grid represents a closer relationship between those inputs. Input "closeness" is determined using methods specific to the type of input being visualized.

The feature map has been used in the visualization of query matches to a document database [15]. Specifically, it was used for query matches based on statistics, probability theory, etc., where some measure of the "closeness" of the items returned could be applied. The work even suggested the feature map as a natural counterpart to any such set of low-precision, "fuzzy" input where the user must be able to easily browse their relationships and relevance.

One of the feature map's advantages is that it can represent dynamic structural relationships among a set of inputs. In our case, however, we have a Bayesian model whose structure currently does not change between situations. Therefore, nodes occupied by the inputs to our map do not significantly change their relative locations. As we shall see, this presents certain advantages to the user in the form of consistency in the interface. For more sophisticated networks whose structure changes as new variables and relationships are learned, the feature map algorithm could prove especially useful.

Figure 4 (a and b) shows our feature map variant for two events using the same calendar day of the previous example as input. The entry is confined to a rectangular shape for integration into a standard calendar program. The rightmost sections of the entries indicate the event title and the attendance likelihood in a vertical bar graph identical to our first prototype.

The boxes to the left of the bar graph comprise the feature map. Subdivisions within the map represent variables in the Bayesian model of Figure 2, and are organized to indicate the most important influences in the network and their relationships. Therefore, only the most influential variables are represented in the map. Horizontal adjacencies between

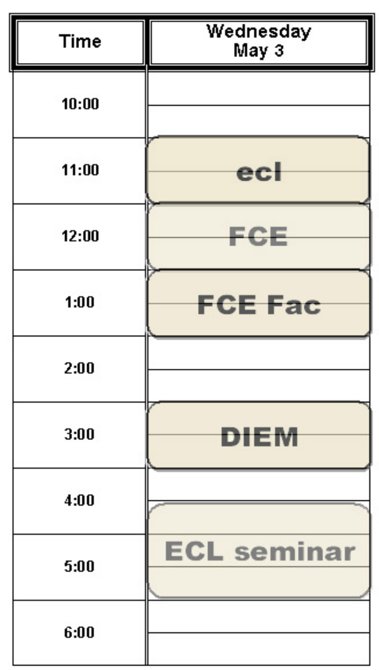

Figure 3: A prototype that uses transparency

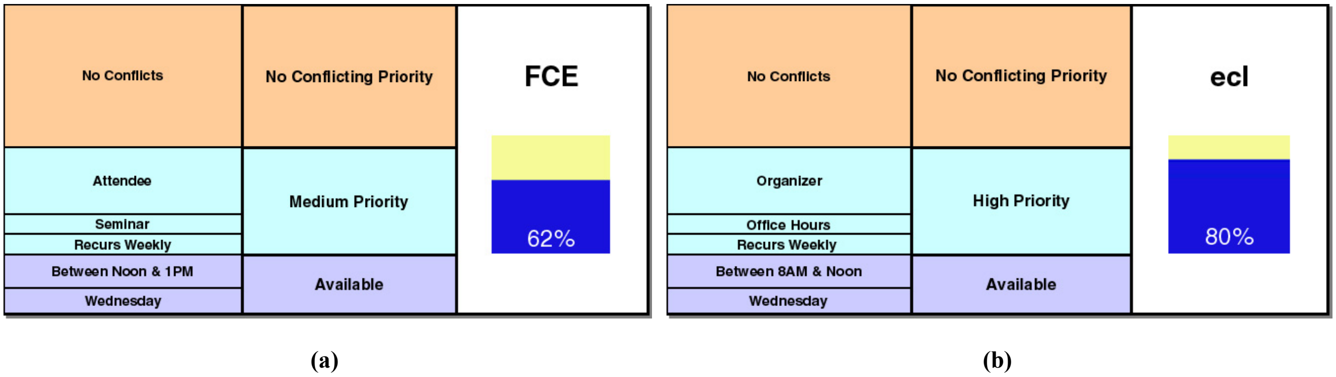

Figure 4: Visualizations of the first two calendar entries used in Figures 2 and 3. Larger boxes indicate greater influence over horizontally adjacent variables. The rightmost boxes, adjacent to the title and bar graph, indicate the most direct influence over the final prediction of attendance. Note how the person's roles in the event ("Organizer" in Figure 4a and "Attendee" in Figure 4b) seem to be the main reasons for the difference in attendance likelihood between the two entries.

boxes indicate relationships between variables, and are color-coded for clarity. Although true feature maps also make use of the vertical dimension to depict relationships, the structure of our network is sufficiently simple that we do not use this dimension. Larger boxes indicate greater influence on related variables. The Netica software calculates this degree of influence by performing a sensitivity analysis using an entropy reduction metric. Variables that exert the most direct influence on the final prediction are further to the right.

As an example, suppose we wish to determine why the likelihood of attendance for the event named "FCE" is lower than that of the event named "ecl". The largest direct influence on the system's prediction seems to be the priority of any conflicting events, indicated by the top row of boxes in the map. Since there are no conflicting events for either entry, we look to the next highest influence, which is the priority of the event itself, occupying the middle section of the map. We see that "ecl" is most likely a high priority, while "FCE" is a medium priority. To find out more, we look at the variables influencing the priority, which are the identically colored boxes to its left. We see that the calendar's owner is the organizer of the "ecl" event, but only an attendee of the "FCE" event. Therefore, we conclude that the owner's role in the event is the most likely reason for the disparity between the two entries.

This visualization is also useful for diagnostic purposes. If a person browsing this information is skeptical of the results returned, an examination of the map can reveal incongruities between the system and user models of the situation. For example, an earlier version of the prototype depicted an event with the word "seminar" in its title. Upon examining the feature map, the calendar's owner observed that although that particular event was technically a seminar, it should be treated as a course. We then modified our text-parsing component (discussed later) to account for this event.

Evaluation

Although no formal evaluations have been conducted on these techniques, informal assessments by students and faculty within the department raised a number of issues, prompting several refinements to the prototypes introduced in this work. Users were able to perform comparisons of entries visualized using our modified feature map, identifying the key differences that produced the final prediction. Larger subdivisions in the map were correctly identified as heavier influences toward predictions, and the color-coded layout cued them in to relationships in the network. However, the exact relationship between the influences represented in the map and the final scalar output was not clear. Users seemed to want the colors used in the map to be incorporated into the bar graph on the right in some additive fashion. Unfortunately, interrelated variables in the network and negative influences conspire to make this type of association difficult. Nevertheless, we feel that this technique, and its future refinements, represents an important area of research, as more intelligent components are included in ubiquitous computing systems.

For our semi-transparent prototype, the use of a continuous mapping from attendance probability to opacity made it difficult to distinguish significant differences between entries. We therefore discretized the mapping to emphasize those differences.

System Components

The system is currently implemented in three components. The first is a web-based "attendance diary" program that presents the calendar's owner with a checklist of the day's scheduled events. The owner checks which events were attended and which were not, and submits the list. The diary is implemented as a CGI script, and calendar data is obtained by parsing the owner's Palm datebook data file.

There were several reasons behind our choice of the Palm datebook. First, unlike many organizations, our academic department does not have a standard groupware calendar system such as Microsoft Outlook or Netscape Calendar available to all students and faculty. As such, the most commonly used calendar system turns out to be the Palm datebook. Second, while the datebook provides a rather impoverished set of event attributes, the learning evidence generated from datebook data proved to be sufficient in obtaining sensible predictions.

A second module, running in the background, checks for new submissions from the attendance diary, converts them into learning evidence for the Bayesian network, and saves a new network with updated probabilities that reflect the new evidence. Network learning is performed through the Netica API.

Many of the Palm datebook fields have straightforward mappings to the variables in our Bayesian model. Several variables, however, such as User Role and Location, do not have equivalent fields in the Palm datebook. Therefore, we perform some rudimentary text parsing of the event title in an attempt to extract evidence for these variables. For instance, if a particular course number is found, the location is easily determined from the school course directory. The calendar owner's status as faculty or student establishes his or her role in that course as either attendee or organizer. Other proper names and keywords are also used to provide similar mappings. Although the mappings currently exist as a static list of keyword-variable pairs, a future refinement of the system will be to learn new mappings either by explicit user descriptions or by observing the user and forming a rule base in a manner similar to the CAP system.

A third component, implemented as a C++ class, takes a given calendar event, sets its attributes as evidence to the Bayesian network, and performs probabilistic inference to arrive at a likelihood of attendance for the event. Again, the Netica API is used to perform network manipulation and inference. Other nodes of the network can be examined as well, and one can even determine the variables that are exerting the most influence on a given variable. This class is intended to be available to any application programmer who wishes to incorporate probabilistic calendar data.

A browsable calendar for one of our faculty members is currently deployed as a CGI-based web page accessible to members of our department. Currently, it only uses the bar graph visualization technique, as our other methods do not yet lend themselves well to HTML description. Updates to the calendar are performed via network synchronization of the faculty member's Pilot with the workstation currently running the Bayesian network software.

CONCLUSIONS AND FUTURE WORK

We have presented Ambush, a probabilistic calendar that provides informed predictions of attendance at future events. We have created a Bayesian model of user attendance habits to supply these predictions. We have demonstrated the importance of calendar information in its traditional role as a stand-alone means of supporting individual and group work, and also in its more novel capacity as a provider of contextual data to a growing number of wearable and ubiquitous computing applications. We feel that a probabilistic calendar is of great importance to applications attempting to use such information to determine user availability or location.

Toward our own specific research goal of using calendars and other contextual information to support informal communication in the workplace, we have presented several techniques for visualizing the probabilistic output of our calendar. These techniques included a simple mapping of attendance likelihood to a bar graph or transparency level, as well as a more sophisticated approach that attempts to let users leverage the readability of Bayesian networks to form a better conceptual model of the system.

A probabilistic calendar could serve as a useful sensor in a wide range of applications, ranging from communications systems that consider the availability of the user to recommender systems that make use of the user's current working context. To enable fast integration of our system into such applications, we are pursuing its inclusion in the Context Toolkit [21], a framework for building context-aware applications.

We recognize that the attendance diary is an additional burden on the user. Efforts are underway at Georgia Tech to provide location tracking to both faculty and students. This information could be used in conjunction with the calendar to automatically check attendance by correlating calendar entries with the current location of the owner and possibly any other attendees listed. We hope to deploy our system on a larger scale to collect more attendance data, thereby improving the model's predictions. Given that the extra work involved in having each user maintain a diary would make large-scale deployment impractical, the elimination of this component is a high priority.

Mitchell's CAP system [18] exhibited poor performance during the large schedule disruptions caused by the semester boundaries in the academic year. We expect some performance degradation as well, since user availability will change as classes are shifted to new dates and times. Since the learned probabilities from the previous semester associate availability with event dates and times, we can consider mapping these probabilities to the dates and times of new events on next semester's schedule. This could at least establish a baseline and possibly shorten the learning time required to produce sound predictions about attendance.

ACKNOWLEDGMENTS

The authors would like to thank the Everyday Computing Lab, the Future Computing Environments group, and Eric Horvitz. We also thank the National Science Foundation for support.

REFERENCES

1. Beard, D., Palaniappan, M., Humm, A., Banks, D., Nair, A., and Shan, Y. A visual calendar for scheduling group meetings. *Proceedings of CSCW 1990*, Pages 279 – 290.
2. Birnbaum, L., Horvitz, E., Kurlander, D., Lieberman, H., Marks, J., Roth, S., Compelling intelligent user interfaces: how much AI? *Proceedings of IUI '97*, 1997.
3. Conati, C., Gertner, A., VanLehn, K., and Druzdzel, M. Online student modeling for coached problem solving using Bayesian networks. *Proceedings of the Sixth International Conference on User Modeling*, 1997, Pages 231-242.
4. Druzdzel, M. Qualitative Verbal Explanations in Bayesian Belief Networks. *Artificial Intelligence and Simulation of Behavior Quarterly* 94, 1996, Pages 43-54.
5. Ehrlich, S. Social and psychological factors influencing the design of office communication systems. *Proceed-*

ings of the ACM CHI+GI'87 Conference, 1987, Pages 323-329.

6. Fish, R., Kraut, R., Root, R. and Rice, R. Evaluating video as a technology for informal communication; *Proceedings of CHI'92*, 1992, Pages 37 – 48.
7. Goodman, G. and Abel, M. Communication and collaboration: Facilitating cooperative work through communication. *Office: Technology and People*, **3** (2), 1987, Pages 129-146.
8. Harrison, B., Kurtenbach, G., Vicente, K. An experimental evaluation of transparent user interface tools and information content. *Proceedings of UIST '95*, 1995, Pages 81-90.
9. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. *Proceedings of UAI 1998*.
10. Horvitz, E., Jacobs, A., and Hovel, D. Attention-sensitive alerting. *Proceedings of UAI '99,* July 1999, Pages 305-313.
11. Horvitz, E. and Paek, T. A Computational Architecture for Conversation. *Proceedings of the Seventh International Conference on User Modeling*, 1999, Pages 201-210.
12. Kohonen, T. *Self-organization and associate memory.* Third Ed. Springer-Verlag, New York, 1989.
13. Kozierok, R. and Maes, P. A learning interface agent for scheduling meetings. *Proceedings of IUI '93*, 1993, Pages 81-88.
14. Kraut, R.E., Fish, R.S., Root, R.W., and Chalfonte, B.L. Informal communication in organizations: Form, function, and technology. In S. Oskamp & S. Spacapan (Eds), *The Claremont Symposium on Applied Social Psychology*, 1990, Pages 145-199.
15. Lin, X. Visualization for the Document Space. In Card, S., MacKinlay, J., and Schniederman, B.(Eds), *Readings in Information Visualization*, 1999, Pages 432-439.
16. Madigan, D., Mosurski, K., and Almond, R. Graphical Explanation in Belief Networks. Journal of Computational Graphics and Statistics 6(2), 1996, Pages 160-181.
17. Marx, M. and Schmandt, C. CLUES: dynamic personalized message filtering. *Proceedings of CSCW 1996*, Pages 113 – 121.
18. Mitchell, T. M., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. Experience with a learning personal assistant. *Communications of the ACM* 37, 7 (Jul. 1994), Pages 80 – 91.
19. Mosier, J. and Tammaro, S. When are group scheduling tools useful? *CSCW: The Journal of Collaborative Computing*, 6, 1997, Pages 53-70.
20. Palen, L. Social, individual and technological issues for groupware calendar systems. *Proceedings of CHI '99*, 1999, Pages 17 – 24.
21. Salber, D., Dey, A., Abowd, G. The Context Toolkit: aiding the development of context-enabled applications. *Proceedings of CHI '99,* 1999, Pages 434-442.
22. Suermondt, H.J. and Cooper, G.F. An Evaluation of Explanations of Probabilistic Inference. *Computers and Biomedical Research* 26, 1993, Pages 242-254.
23. Yan, H. and Selker, T. Context-aware office assistant. *Proceedings of IUI 2000.*