# Recognizing and Using Goals in Event Management

**Dustin Smith**
MIT Media Lab
20 Ames Street
Cambridge, MA 02139
dustin@media.mit.edu

**Henry Lieberman**
MIT Media Lab
20 Ames Street
Cambridge, MA 02139
lieber@media.mit.edu

Copyright is held by the author/owner(s).
*CHI 2009*, April 4 – 9, 2009, Boston, MA, USA
ACM 978-1-60558-246-7/09/04.

## Abstract

Event management involves planning when, where and how events should occur, making sure the event's prerequisites are satisfied, and developing contingencies for when things go wrong. Conventional calendar and project management tools, however, only record and visualize explicit human decisions regarding event specifics.

We present *Event Minder*, a calendar program that takes into account the *goals* for which the events are scheduled. Users can input descriptions of events in natural language, mixing high-level objectives, concrete time and place decisions, and omit "obvious" common sense details. A commonsense knowledge base provides sensible defaults, and machine learning refines these defaults with experience. We can make recommendations for alternative plans, including alternatives that satisfy higher-level goals in different ways as well as those that meet immediate constraints.

## Keywords

PIM, user modeling, calendaring, event management, common sense

## ACM Classification Keywords

H.4.1 Office Automation: Time Management

## Introduction

Our lives are not just collections of isolated events. We present Event Minder, a novel calendaring application that has the following capabilities:

• It can accept unconstrained natural language input (but perhaps not understand everything). Events may be only partially described.

• It relies on the Open Mind Common Sense knowledge base [6], [7] for background knowledge necessary to fill in intelligent defaults for missing details.

• It has knowledge of common personal goals for event management, common ways in which these goals may be accomplished, and what details need to be decided.

• It can connect sequences of related events. If an event occurs at a location other than where the user is, the user must arrive at the event, and return or go to another event.

• It uses machine learning to personalize event details based on past experience. For example, it might assume lunch occurs at noon, but if your lunch time typically occurs earlier or later, it will adjust its default assumption.

• It can present a list of candidate goals for why particular details were chosen, and the user can confirm one or more goals, or introduce new goals.

• It can support dynamic *replanning*, in the case that a particular detail becomes infeasible. It can suggest alternative plans that satisfy the user's likely goals. If the user's goal was to discuss a particular project, the time and place of the meeting might be changeable, but not the participant list for the event.

• It integrates Web resources such as real-time restaurant reviews, subway schedules, etc. seamlessly into the event scheduling process.


**figure 1.** An example of the Event Minder interface.

The bidirectional inference between statements of goals and concrete event details is Event Minder's most unique contribution. We think that, in many applications, the most flexible and helpful interaction between humans and computers occurs when the person is not constrained to make decisions exclusively at either a high level or a low level, but can freely mix requests at many levels. The responsibility then is on the system to understand how high-level goals and lower level decisions relate.

Requiring complete low-level specifications, as today's calendars do, gets tedious. Many AI systems, though, err in the opposite direction. They require the user to think too abstractly, often in predefined categories or abstract ontologies. Often, the user would prefer simply to specify uncontentious details directly, without having the system get in the way. *Event Minder*'s approach provides the best of both worlds.

*Event Minder* operates primarily in the domain of planning restaurant meals, including transportation to and from events. It connects with the CitySearch database to provide data about local restaurants. With some additional knowledge collection and modeling efforts, it can be extended into other personal event planning domains. We are also planning a mobile version on a phone or PDA platform, tracking the user's location in real time.

*Parsing Natural language input*

*Event Minder* takes input in natural language, which is often the easiest and most accessible way for users to express their needs. Parsing English event descriptions is tricky because there are many ways someone could describe the same event; for example: "lunch", "lunch with joe tomorrow", or "lunch with Joe at Legal Sea Food".

We restrict ourselves to answering questions that are relevant to our application--namely those pieces of knowledge that are related to finding the best *when, where* and *how* the event will take place. These variables are represented with a default event frame that contains predefined event slots (see left margin).

When the user enters a description, we look to see which slots have been specified. Identifying the correct components of a semantic frame is known as *semantic role labeling*, and is challenging because there are ambiguities in aligning values to slots, and in identifying the correct slot instance. For example in "Lunch at Morton's" it is unclear whether "Morton's" refers to an upscale steakhouse, or to your friend, Morton's house.

A likelihood score is computed for each possible interpretation. We keep a dictionary of local locations, attendee names, and event types and compute string similarity. Dates are parsed using a rule-based date description parser that either accepts or rejects the input string. Likelihood scores are computed and the best slot-alignment is selected.

*Filling in missing specifications*

With the event description, "Lunch tomorrow with Joe", a lot was left unsaid. Where is Lunch? What time will it start? From the available details about the event, other details are inferred: Lunch is a meal that typically occurs in around noon, but we don't know when exactly or for how long.

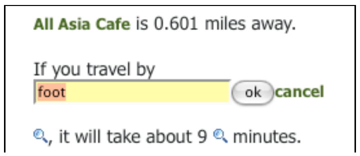


**figure 3.** All of the event details can be edited by the user. Clicking on a detail will expose a hidden form box. One can change the times, dates, venues, participants, etc. The system has guessed that since the restaurant is at such a short distance, the user will likely walk. If the transportation mode is edited to "drive", the travel time is recomputed accordingly.

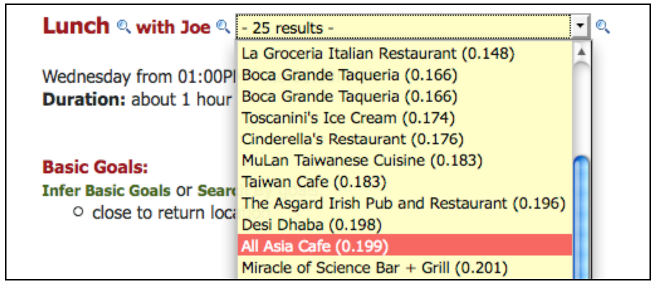


**figure 4.** A list of restaurants near the inferred origin location is presented in (along with their distance in miles).

The user selects "All Asia Café", and *Event Minder* fills in default values that are contingent upon the location, such as the time it takes to travel to the restaurant.

*Infer by Goals and Search by Goals*
In *Event Minder*, there is an operation "Infer or Search by Goals" that is useful for situations where the user gets off-schedule or goals cannot be fulfilled for some reason. Suppose All Asia is closed on the particular day chosen? If you're late, our knowledge about transportation modes can replace a cheap but slow bus ride with a fast but expensive taxi ride.

If we choose to Infer Goals for the All Asia example, a list of possible goals is computed. Since All Asia has a liquor license, "Drink booze" is a possible goal.


**figure 5.** A list of restaurants near the inferred origin location is presented in (along with their distance in miles).

Here, though, we have an unusual combination. If we check "hear music" (the restaurant in question often features live music), and also "eat Asian food", that narrows down the possibilities considerably. Other than our original choice, All Asia, the restaurant Kapow is the only one in Boston to feature Asian food, serve alcohol, and present live music (see figure 6 on the following page).

How is this done? We use knowledge that takes the form as a relation between event details and goals. Goals are represented as a hierarchical list (although a flat list is displayed in figure 5). Event details take different representational forms: Restaurants are represented as sets of binary features.

When users interact with goals, they are interacting with hidden relations that affect properties of the event. This hidden knowledge takes the form of relations between event details and goals. In the case of locations, goals are mapped to Boolean combinations of location features (for restaurants, in this case):

Be healthy = Seafood $\vee$ Vegetarian $\vee$ Health food
Avoid eating meat = Vegetarian $\vee$ $\neg$ Meat

This way we infer the user's goals from specific restaurants they have selected, and, when the user specifies goals, we construct ad hoc categories of restaurants by combining features.

Over time, constraints between goals can be used to reduce the hypothesis space in inductive learning. For example, if the user visits both a sushi restaurant and a vegetarian restaurant, the system could infer that his or her goal is to "Be healthy", not "Avoid eating meat". This would allow the system to suggest other healthy restaurants, including those that serve meat.

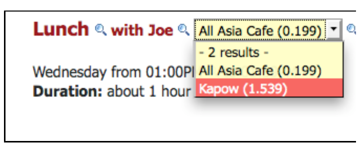


**figure 6.** The two restaurants in the search results that match the ad hoc category constructed from the user's goals.

## Related Work

Work on calendar programs, meeting schedulers, project management software, and related time management assistance is too numerous to review in detail. Most do not attempt any serious semantic interpretation of the content of calendar entries, but a few do try to connect event semantics to assist the user in time and place decisions.

Mueller's Commonsense Calendar [9] is the only reference that explicitly tries to use common sense knowledge to aid calendar management. Its primary function is "sanity checking" entries to avoid conflicting situations such as planning to take a vegetarian to a steak house. The present work is more concerned with the elaboration of details in normal operation rather than detection of exceptional cases.

Previous work on filling in missing information in calendar entries has focused exclusively on event times. Mitchell et al.'s CAP [8] is a calendar learning agent that infers meeting time, duration and other parameters from examples, Gervasio et al [4] advance this with active learning. Berry et al [1] address the problem of negotiation among calendar agents representing different users. None of these model or try to learn the users goal. But, the use goal knowledge for feature selection in clustering, as we do for location feature vectors, was proposed in Stepp and Michalski [2].

Gil and Ratnakar [5] report on a system that parses natural language specifications of tasks (similar to what we call goals) from "to-do" lists and maps these on to a set of predefined agents. Faulring and Myers [3] extract event details from email, and provide interesting dialog and visualization capabilities in the related RADAR project. Google Calendar has a "Quick Add" feature that allows natural language description of a conventional calendar entry, but we could find no published details about how it works.

Structuring interfaces around user goals has long been a goal of AI-based user interfaces. Roadie [6] allows users to execute multi-step actions among consumer electronic components by stating their goals.

## Future Work

As we expand *Event Minder* to more event domains, it is evident that we need a more expressive representation of goals than a simple generalization lattice. We need a way for users to suggest their *priorities*, because goals, as hard-constraints, are sensitive to the order in which they are applied. For example, people would recognize the event eating dinner at home to have the characteristic goal of "to save money." But, not conditioned upon eating, there are plenty of activities more effective at achieving the goal "to save money." Such a system, if capable, would justify this conclusion: "if you really wanted to save money, you should play Frisbee, watch the sunset or take a walk."

We also plan to soften goal constraints with probabilities. This will allow us to express varying degrees of certainty and to have graded approximate matches against locations in the database. With the goal of eating, "go to a lecture" may be a valid recommended event with a low degree of certainty (often food is found at lectures).

## Conclusion

Essentially, we seek to reproduce some of the helpfulness and flexibility of human assistants in event management tools. We hope we have shown with *Event Minder* that when it comes to event planning, goals really matter.

## Citations


[1] Berry, Pauline M., Cory Albright, Emma Bowring, Ken Conley, Kenneth Nitz, Jonathan P. Pearce, Bart Peintner, Shahin Saadati, Milind Tambe, Tomás Uribe, and Neil Yorke-Smith, Conflict Negotiation Among Personal Calendar Agents, Proceedings AAMAS'06, May 8-12, 2006, Hakodate, Hokkaido, Japan.

[2] Stepp, R.E and Michalski, R.S. Conceptual clustering of structured objects: a goal-oriented approach. *Artificial Intelligence*. 28, 1 (1986), 43-69.

[3] Faulring, A. and Myers, B. A. Enabling Rich Human-Agent Interaction for a Calendar Scheduling Agent. *Ext. Abstracts CHI 2005*, ACM Press (2005).

[4] Gervasio, Melinda T., Michael D. Moffitt, Martha E. Pollack, Joseph M. Taylor, and Tomas E. Uribe., Active Preference Learning for Personalized Calendar Scheduling Assistance, Proceedings of the 2005 International Conference on Intelligent User Interfaces, 2005.

[5] Gil, Y. and Ratnakar V., Automating To-Do Lists for Users: Interpretation of To-Dos for Selecting and Tasking Agents. Proceedings of the Twenty-Third Conference of the Association for the Advancement of Artificial Intelligence (AAAI-08), Chicago, IL, July 2008.

[6] Lieberman, H, and Espinosa J., A Goal-Oriented Interface for Consumer Electronics, 2006 Inernational Conference on Intelligent User Interfaces (IUI-06), Sydney Australia, January 2006.

[7] Liu, H, Lieberman, H and Selker, T. GOOSE: Goal-Oriented Search Engine With Commonsense. Proceedings of the Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference, (AH 02). (2002).

[8] Mitchell, T. M., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. Experience with a learning personal assistant. Communications of the ACM 37, 7 (1994).

[9] Mueller, E. T. A calendar with common sense. In *IUI '00: Proceedings of the 5th international conference on Intelligent User Interfaces* (New York, NY 2000), ACM Press, 198-201.