

A Calendar with Common Sense

Erik T. Mueller

Signiform

2533 Waterside Drive NW

Washington, DC 20008-2820 USA

+1 202 328 0905

erik@signiform.com

ABSTRACT

Digital devices today have little understanding of their real-world context, and as a result they often make stupid mistakes. To improve this situation we are developing a database of world knowledge called ThoughtTreasure at the same time that we develop intelligent applications. In this paper we present one such application, SensiCal, a calendar with a degree of common sense. We discuss the pieces of common sense important in calendar management and present methods for extracting relevant information from calendar items.

Keywords

common sense, calendaring, PIM.

1. INTRODUCTION

In recent years we have seen an explosion in the development and availability of digital devices of all sorts: personal computers, palm devices, wearable computers, digital tags, portable phones, and smart toys. Yet there has been no corresponding explosion in the development and availability of the software that will give these devices the ability to understand their real-world context. These things have no common sense. For example:

- A phone has no idea it can be annoying during a romantic dinner.
- A beeper tries to let you know the battery is low while you are asleep.
- A calendar manager thinks it is fine to arrange a dinner for 6 am.
- A car navigation system fails to check whether the destination, a museum, is even open.

To counteract this trend we are developing ThoughtTreasure (Mueller, 1998a), a treasure-house of commonsense knowledge and reasoning mechanisms. ThoughtTreasure contains on the order of 100,000 pieces of common sense, represented declaratively in some cases and procedurally in others.

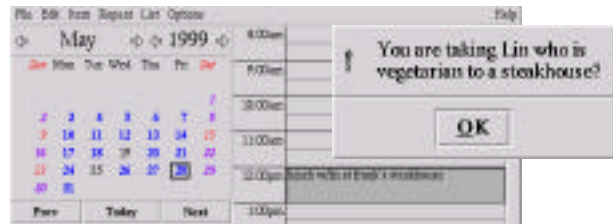


Figure 1: SensiCal points out a problem

Some sample items in ThoughtTreasure are:

- Soda is a drink.
- People have fingernails.
- Dinner lasts about 2 hours.
- The evening extends from about 5 pm to 9 pm.
- A hotel room has a bed, night table, minibar, ...
- A teenager's age is greater than or equal to 13 and less than 20.
- A person has at most one spouse.
- At the end of a phone call, one says goodbye and hangs up.
- Rough synonyms for *food* are: *foodstuffs*, *chow*, ...
- A *cassette* can mean a cassette tape or a cassette recorder.
- Yellow hair is called *blond* hair.

Applications can use ThoughtTreasure to help them understand more about the world in order to prevent stupid mistakes, point out problems, take corrective actions, and provide intelligent defaults. ThoughtTreasure is implemented in C and is available from <http://www.signiform.com>.

Instead of attempting to develop a complete catalog of commonsense knowledge as was attempted in the Cyc project (Lenat, 1995), we are evolving ThoughtTreasure at the same time as the applications that use it. In this way we will grow ThoughtTreasure to be useful.

This paper focuses on one of these applications, a calendar application called SensiCal.

2. SENSICAL

We are developing an intelligent calendar application called SensiCal that provides two improvements over traditional calendar programs. First, it reduces entry time by filling in certain information automatically. Second, it produces warning messages that help the user avoid obvious blunders.

SensiCal runs as an extension to ical (Ghemawat, 1993) or as a batch process on data exported from other calendar programs. SensiCal is written in Tcl and Perl and communicates with ThoughtTreasure using the ThoughtTreasure server protocol (Mueller, 1998b).

Figure 1 shows a sample interaction with SensiCal. The user enters a new calendar item:

lunch w/lin at frank's steakhouse

The application responds with a popup:

You are taking Lin who is vegetarian to a steakhouse?

The user has previously entered the fact that Lin is a vegetarian into a profile for Lin.

How does SensiCal work? Whenever a calendar item is added or modified, SensiCal performs the following steps:

1. Extract relevant information from the text of the calendar item.
2. Fill in missing information.
3. Point out potential problems.

We discuss each of these in turn.

3. INFORMATION EXTRACTION

The first step is to build a machine-understandable representation of the entered calendar item. Starting from the calendar item's text and other fields such as start date/time and end date/time, SensiCal extracts the following information:

- the type of item (such as *todo*, *memo*, *holiday*, *trip*, *meal*, *meeting*, *class*, *talk*, *outing*, *party*, or *TV-show*)
- the participants
 - role (such as *speaker* or *student*)
 - name
 - phone number
- the location
 - venue name (such as the name of the restaurant)
 - type of venue (such as the type of restaurant)
 - Earth coordinates
- more information about the activity (such as the type of meal) or the topic (of a talk or meeting)

For example, from the entered item:

```
Text: lunch w/lin at frank's steakhouse
StartTs: 19990528T120000
EndTs: 19990528T130000
```

SensiCal extracts the additional information:

```
ItemType: meal
MealType: lunch
Participant: Lin
Venue: Frank's Steakhouse
VenueType: steakhouse
```

ThoughtTreasure provides two important resources to assist with the extraction task: a lexicon/ontology and a collection of *text agents* for recognizing various entities in text.

ThoughtTreasure's lexicon/ontology consists of 55,000 English and French words and phrases, 25,000 atomic concepts, and 50,000 assertions on those concepts. The lexical entries and concepts relevant to the calendar application include: human names, geographical place names, holiday names, well-known restaurants, types of restaurants, dietary preferences, common hotels, common stores, types of stores, TV channels, TV shows, meals, classes, and sports.

```
==eating#V place*.z//place* to#R eat#V.z/
===bar.z//pub.z//bar.My/|serve=alcoholic-
beverage|
===café.z//café.My/|serve=coffee|
```

```
===brasserie.z//café#-
restaurant*.z//brasserie.Fy/
===tearoom.z//salon* de#R thé#My/|serve=tea|
===restaurant.z//eatery.Àz/restaurant.My/
===pizzeria.z//pizzeria.Fy/|serve=pizza|
====Domino#S's.z//
====Papa# Gino#S's.z//
====Pizza# Hut*.z//
====Pizzeria# Uno*.z//
====steakhouse.z//steak# house*.z//grill.z/
restaurant* grill*.My//grill.My/|serve=beef|
```

Figure 2: Portion of restaurant lexicon/ontology

Figure 2 shows an abbreviated portion of ThoughtTreasure's lexicon and ontology for restaurants. The level of indentation shows the hierarchy: a restaurant is a type of eating-place and a pizzeria is a type of restaurant. There are two words for the pizzeria concept: *pizzeria* in English (z) and *pizzeria* in French (y).

ThoughtTreasure's text agents are used to recognize and parse entities such as names, times, dates, phone numbers, and prices. Figure 3 shows patterns used by the *restaurant text agent* to recognize and classify restaurants in text.

4. FILLING IN MISSING INFORMATION

In the second step, SensiCal helps the user by filling in missing information and providing intelligent defaults.

SensiCal fills in the length of an appointment automatically. For example, when the user enters a new calendar item:

```
dinner w/Susanna
```

without an end time, the duration is automatically set to 2 hours, the typical length of a dinner. This is done by querying ThoughtTreasure for the typical duration of the activity. Mueller (1999) presents sample Java code for querying ThoughtTreasure for duration.

5. POINTING OUT PROBLEMS

In the third step, SensiCal points out potential problems or violations of common sense to the user. For example, when the user enters:

```
breakfast w/Susanna at City Diner
```

with a time of 3 am, the application responds with:

```
You are eating breakfast at 3 am?
```

The pieces of common sense most important to the calendar application are as follows.

5.1 Space and time

- A person can't be in two places at once.

It is important to know when meetings do not require physical presence, as is the case for videoconferences.

- Allow sufficient time to travel from one location to another.

ThoughtTreasure plans and estimates the travel time of trips from one location to another using various modes of transportation such as walking, driving, taking the subway, and flying.

- People usually live near the organizations they work for.

```

X BBQ | X Barbecue | X Ribs → barbecue-
restaurant
X Noodle | Ginger X | Golden X | Hunan X | X
Hunan
  | Sichuan X | Szechuan X | X Shanghai | X
Garden
  | X Palace → Chinese-restaurant
X Deli → deli-restaurant
Le X | Les X | Au X | Chez X | Du X →
French-restaurant
Bombay X | Delhi X | India X | X India | Indian
X
  | X of India → Indian-restaurant
Al X | Caffe X | Cucina X | Da X | Il X
  | Ristorante X | Trattoria X →
Italian-restaurant
X Sushi → Japanese-restaurant
Baja X | Casa X | Mi X → Mexican-restaurant
X's Pizza | X Pizzeria | Pizzeria X → pizzeria
X Seafood | X Sea Food | X Sea Foods | Pier X
  | X Pier | X Fish House | X Fish | X Island
  | X Ocean | Ocean X | Lobster X | X Lobster →
seafood-restaurant
El X → Spanish-restaurant
X's Steak House | X Steakhouse | X Steak | X
House
  | X Grill | X Grille → steakhouse
Bangkok X | Siam X | X of Siam | Thai X | X Thai
  | Thailand X | X Thailand → Thai-restaurant
X Vegetarian | Vegetarian X →
vegetarian-restaurant
Saigon X | X Saigon | Pho X →
Vietnamese-restaurant

```

Figure 3: Restaurant classification patterns

5.2 Venues

- You can't visit a place that's not open.
- Restaurants do not generally serve dinner after 11 pm.
- Museums are often closed on Mondays. (etc. for other places)

5.3 Activities

- People typically work during the day and sleep at night.
- People do not usually attend business meetings on holidays.
- Lunch is eaten around noon for about an hour. (etc. for other meals and activities)

5.4 Settings

- People eat in the dining room or kitchen and in restaurants. (etc. for other activities)
- Phones are found in offices, living rooms, kitchens, bedrooms, and gas stations. (etc. for various objects)
- People purchase items in stores that sell those items.
- Vegetables are sold in grocery stores. (etc. for other products)

5.5 Food

- Avoid restaurants that serve mostly food you don't eat.
- Vegetarians don't eat meat. (etc. for other dietary preferences such as low-fat)

- A steak house serves steak. (etc. for other types of restaurants)
- Steak is meat. (plus a complete food taxonomy)
- To cook a recipe you need the ingredients.

5.6 Social

- People celebrate holidays with their families.
- People support their friends by going to their performances and talks.
- People show up at events organized by their friends and family.

Some of these pieces of common sense are contained in ThoughtTreasure. Others are hardcoded into SensiCal.

ThoughtTreasure uses multiple representation schemes for common sense: grids, assertions, and procedures.

ThoughtTreasure's grids provide information about where objects are typically found. Currently 28 grids are defined. Here is an abbreviated version of a grid that represents a hotel room:

```

==Park-Plaza-1E//
wwwwwwwwwwww      b:bed
wbbbbb            mw      d:lockable-door
wbbbbb            w       m:minibar
wx                Zw      w:wall
wwwwwwdddwww      x:phone
w                  x:night-table
wwwwwwwwwwww      Z.wd:hotel-room

```

The key on the right specifies which objects are associated with which characters: w represents a wall and m represents a minibar. Contiguous characters in the grid represent single objects. (There is only one bed, which occupies all the cells containing a b.)

The hotel room is defined by filling a region of the grid: Z.wd specifies to fill all cells reachable from the Z character without going through a w (wall) or a d (door). Mueller (1999) presents sample Java code for using ThoughtTreasure grids to determine the typical locations of objects.

ThoughtTreasure's assertions represent facts and inference rules such as the typical time of day, duration, and location of an activity, the taxonomy of foods, and dietary preferences.

ThoughtTreasure's procedures are used for simulating the behavior of people attempting to achieve a goal, simulating the behavior of objects and devices, and for determining travel time.

SensiCal uses ThoughtTreasure as a resource when examining calendar items for potential problems. When checking that a restaurant serves food that satisfies the dietary preferences of a participant, SensiCal queries ThoughtTreasure for what the restaurant serves and what foods someone with those dietary preferences avoids.

Information about the dietary preferences of a person is stored in a SensiCal profile. This profile also includes other information about the person useful for commonsense inferring: relationship to the user, birthday, spouse, family members, friends, hobbies, and occupation.

When checking for sufficient travel time, SensiCal invokes ThoughtTreasure's trip planner:

1. Form a list of all participants (including the user) from the calendar.

2. For each participant, for each calendar item for that participant in chronological order:
 - Plan a trip from the location of the previous item to the location of this item.
 - If the trip duration is much greater than the start time of this item minus the end time of the previous item, point out the problem to the user.

SensiCal also takes into account calendar items of type trip such as:

```
Annie in town
to Chicago
back to Boston
```

For each trip, it considers calendar items that overlap the trip and points out a problem if there is insufficient travel time. When the user enters:

```
computing orientation N42 demo center
```

on a day when the user is out of town, SensiCal responds with:
You'll be in Chicago.

6. FUTURE WORK

In future work we will develop ThoughtTreasure's commonsense knowledge relevant to the calendar application and improve SensiCal's ability to exploit this knowledge. We will move some of the functionality hardcoded in SensiCal back into ThoughtTreasure so it can be used by other applications.

Evaluation will be necessary to determine whether and how often the messages and defaults produced by SensiCal are helpful to the user. It should be easy to override or eliminate certain suggestions and defaults, and introduce exceptions to commonsense rules. For example, it may be the case that City Diner is open at 3 am, and that the user actually does have an appointment there at that time.

SensiCal should be able to learn from previously entered calendar items. For example, the application could notice that whenever the user has lunch with Lin, the user goes to a vegetarian restaurant. This would eliminate the need to add this information manually to the profile for Lin. Since the user often enters durations, typical durations of activities could be learned. Travel times could also be learned from the times of reminder alarms associated with items. Kozierok and Maes (1993) have built a calendar agent that learns a user's preferred scheduling rules.

7. CONCLUSION

Commonsense knowledge enables a calendar program to fill in likely defaults and prevent common mistakes in the absence of

more detailed information. As smart devices become more connected with their users, the world, and each other, we can envision a number of other scenarios in which common sense will be useful:

- A wearable audio platform (such as the one described by Sawhney & Schmandt, 1999) tells you when a nearby store probably carries an item on your todo list.
- Stereos turn down their volumes when you become involved in a conversation.
- Elevators guess whether you want to go down or up.
- Devices beep to remind you to remove them before going swimming.
- Heart rate monitors know when you are exercising and not simply excited.
- T-shirts warn you when you go to work wearing them on the day of a client meeting.

8. ACKNOWLEDGMENTS

This work was supported by Signiform and the MIT Media Laboratory.

9. REFERENCES

1. Ghemawat, Sanjay. (1993). *ical: An X based calendar program* (Online). Available: http://www.research.digital.com/SRC/personal/Sanjay_Ghemawat/ical/doc.html Palo Alto, CA: Digital Equipment Corporation.
2. Kozierok, Robyn, & Maes, Pattie (1993). A learning interface agent for scheduling meetings. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces* (pp. 81-88). New York: ACM.
3. Lenat, Douglas B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 33-48.
4. Mueller, Erik T. (1998a). *Natural language processing with ThoughtTreasure*. New York: Signiform.
5. Mueller, Erik T. (1998b). *ThoughtTreasure Server Protocol* (Online). Available: <http://www.signiform.com/tt/htm/ttsp.htm>
6. Mueller, Erik T. (1999). *Adding common sense to applications using ThoughtTreasure's Java-based client API* (Online). Available: <http://www.signiform.com/tt/htm/addsense.htm>
7. Sawhney, Nitin, & Schmandt, Chris. (1999). Nomadic Radio: Scaleable and contextual notification for wearable audio messaging. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM.