Wen Dong

Project Exercises

Week_2

I propose to implement a calendar which takes the form of a pet (software application) residing in a computer. This calendar can communicate with people in the form of an msn messenger or a yahoo messenger. It accumumates people's various information through cellphone, badge, os log files, etc. It also prompts users for events (calender), summarizes people's statistics (weblog), and scans for people's behaviour changes and compares people's behaviour with those of other people by using a hidden Markov model. It has different behaviours according to people's different state.

This application is relational because the bot adjust its behaviors according to the user. It is relational also because it connects different people together. By tweaking the look and feel, I can make it sticky. It is also helpful in that it fosters good habits.

When everything goes fine, I will consider port the implementation into some hardware (cellphones or sbc's)

Week_3

* What aspects of human-human and human-animal relationships do you want to use ?

The calendar and the user are intimate friends. They show affections to each other.

When the user changes the schedules (i.e. late for working, not attending an approintment) or when the user's schedule is different from others' the calendar should prompt.

The calendar should also learn new things and do something for the user based on the pattern of the user. (for example, whenever the user comes to the office, he opens matlab for programming. He also launches Exceed and connects to some specific linux box, etc.)

* Why does this aplication benefit from creating a relationship ?

The purpose of the application is to help the user to form a good habit. Thus making a user sticking to this application is important. If the user won't stick the the application, it certainly cannot attain its goal.

From the machine's point of view, it should cooperate will with the user. It should not be boring --- thus maybe it might need upgrading through the network so it has new functionalities to attract users. Understanding natuaral language certainly helps.

Understanding visual and audio signals and other physiological information should also help. But these requirements and comments certainly make software design harder.

Elisa-like bot might be a good first step.

* How do you establish the relationship ?

As I mentioned in the previous question : frequent upgrading, multi-sensor, natural language, etc.

Question: How does a people guarentee he can keep relationship with (most) other people ? Is there a good answer ?

* What will need to happen to maintain the relationship over time ?

People might get bored with the application. The human-machine might not able to communicate well. Thus ad-hoc patches are necessery.

Week_4

* Detailed ideas about measuring and evaluating projects.

The interaction is in terms of text. I would like to add speech if possible. The features for measuring and evaluating are all from the media The interaction uses. The features can also help the machine to improve its behavior.

Useful features I can think of:

* Accumulated time people interact with the agent.

* Does the interaction effective in terms of the percentage when designed goals are fulfilled ?

* Does people change their behavior patterns ?

* Is the software more easy to use in terms of average number of keystrokes per purpose ?