Jason He, Brad Orego, Luke Petrinovic 12/15/10
CSC 212 Final Project Summary

The Digital Secretary: Automated Google Calendar Event Entry

Google Calendar, like many computer programs, was designed to mirror its real life counterpart. Like a paper calendar, it is a large sheet on which you put events and plans. Also like a paper calendar, it is designed to be checked by the user by going to it and scrolling through it to the correct date, just like they would flip through the pages of a calendar hanging on the back of their door. But with emulating the old, low tech system of paper calendars as the base for their design, the designers of Google Calendars did not take into account many new possibilities. Computers can support much more complexity than paper, and if we use the needs of the user, rather than the already existing interface with all its flaws, we can find many new ways to innovate and improve Google Calendar.

This was our motivation behind The Digital Secretary: getting Google Calendar to interface with the needs and will of the user rather than just be another tool or instrument to be operated by them. Namely, we see the act of having to check your calendar for availability when you receive an online invite to be a needless waste of time and energy. It requires accessing your Google Calendar and scrolling through to the right date. Since all the information stored in your calendar is easily accessible to any program, we created The Digital Secretary to access it with the information of a given event, and then automatically tell you whether or not you have anything already scheduled in that time slot. The Digital Secretary will then give you the option of scheduling it, or scheduling it over an event that is already planned.

Through our user and task analyses we found that most peoples' scheduling methods can be complex and unreliable. Many people used paper formats (agenda, calendar, note/scrap sheet), which they would often lose or forget about. Those that did use digital calendars, like Google Calendar, used it the same way that they would a regular calendar or planner: they would always be checking and organizing it. Most of these people were actually quite pleased with Google Calendar, because it seemed to them to be much easier and more efficient than paper calendars. These people are our target group, while they use Google Calendar they still go through needless, time wasting tasks almost without knowing it, thinking that they are being efficient while really they are still emulating the same flaws of paper calendars. With our program, working mainly behind-the-scenes and having as little interaction with the user as possible, we can correct these flaws.

In designing The Digital Secretary we had to compete with the task of checking your Google Calendar. We had to make this small task smaller, quicker, and less annoying. We decided that the actual user interface should just be a small pop up that relays only the most necessary information and performs the small task of scheduling, or rescheduling an event. We decided that the most useful time and place for this window would directly after someone decided to schedule an event by clicking the "Add to Google Calendar" button (or hyperlink) that can be added to invitations. The window would inform the user of their availability in that time slot of their schedule and give them the option to schedule. A link to their Google Calendar is provided for convenience. Once the user has been informed of their availability and chooses whether or not to schedule the event then the task is done, and they have no need to go to their Google Calendar.

The way the system works is through interface with Google's Calendar Data API. Essentially, the system authenticates with a user if they are not currently signed in, and from there, runs a by-date search of the calendar data, checking a user's calendar for the exact date and time in question. If there are no answers, it reports that the time is free, and offers the user the ability to add the event via pop-up dialogue box. If it is taken, the user gets a different box, notifying them of what event is in conflict, and presenting them with the option to cancel the conflicting event. In both cases, if the user chooses, the system automatically generates an event with the data harvested from Google's invite URL. Google's Calendar data is based upon XML, which creates the possibility of embedding a calendar event's data into the URL, as opposed to using a unique identifier.

From a code standpoint, the Javascript embedded in index.html first searches the website's DOM for any links that fit the pattern of a Google Calendar invite. If one is found, what happens is that link is re-directed from the original address to the Javascript function. This function then harvests the data it took from the link URL to extract all of the information about the event, which can be passed to the look-up and to the creation functions. test.js is the file that actually searches the user's calendar for events (using CalendarEventQuery), and update.js creates the event (making a CalendarEventEntry object with the data provided by the link). The operation is all completely behind the scenes, and completes in a matter of seconds, vastly increasing throughput when compared to the current system.

Our project is an attempt to solve a problem that most people do not even realize they have, and to interface Google Calendar with the user in a way that, for the most part, they won't notice. We think that this is the real aim of Human Computer Interaction; not just to have the user understand how they are using the computer but to have the computer understand how it is being used, and to react accordingly.