

MAS 965

Relational Machines

Final Project Report

Sajid Sadi

Goals & Overview

The UbER-Badge personalization system, dubbed GuideStar, was designed with consideration to the human-machine relationship that could develop between the system and its users, the sponsors. This text will detail the various aspects of the system. Since the system was highly designed from the beginning, much of it has already been documented previously in high detail, and I will refer to those records with minimal summaries instead of including the extended discussions into this paper. The design of the system was a collaborative effort between many members from 3 research groups as well as the sponsor offices. Often times, design decisions were made in focus groups or through intensive discussions of issues with the entire team, and some portions of this paper refer to those collective decisions in the plural tense.

Interaction

The actual badge was, in the final version, used mostly for location and identification of sponsors. While we had originally planned to use the badge itself as the input device, it was later found that due to the new shorter lanyards used, the sponsors could not see the buttons, thus leading to confusing instructions for use. We thus chose to use the mouse to trigger interactions and keyboard for text input for certain functions. While it was expected that a keyboard would be used, it transpired that the mouse was considerably easier to use in the GUI design, and it was used extensively.

When a badge wearer approached the kiosk, the kiosk immediately greeted the potential user by name, and showed immediately relevant information such as queued messages and available functionality. This was designed to attract the user to the system. The coloration and text sizes were intentionally chosen to attract visual focus at a low level to the name. The additional information was also similarly highlighted, but written in a smaller font intended to be read from a closer distance. This served a dual function of partially obscuring more personal information, and drawing the user closer to the kiosk, which then reduces the barrier to entry (since the user has then already committed attention in order to approach the system). The primary focus of this first screen lay in attempting to “hook” the user into using the system.

From this point, several options were made available, including finding a person, sending messages, receiving messages, and getting recommendations. The intent of the added functionality was to, as again, give the users a number of avenues for interacting with the system, thus increasing the chances that they would become interested and use some part of it. I will focus primarily on the personalization aspect, since that is where most of the relational aspects of the system reside.

The entry of topics for personalization itself is done through a sort of game. In the game, words appear from the right side of the screen, and move towards the left. The more one moves towards the words, the larger they become. The horizontal distance also increases speed in order to quickly reach a word. The goal of the game is to hit the words that are the most interesting. The game is designed to be both fast paced and somewhat “addictive.” One of the first requirements was that the game should last less than a minute, and speed was crucial. While it is possible to play the game slowly, it is designed to get the user to compete and play faster, thus both reducing total time needed for the personalization, and the engagement with the system. While I originally intended to use the speed of motion to calculate an interest metric, experimentation showed that the speed was not necessarily a good indicator, since in tests individuals also took more time when several interesting choices were present, or when no interesting choices were present. Since this effect appeared even in users from the lab already well versed in the lab’s terminology, it was certain that the effects would only be magnified in naïve users, and I abandoned the effort for the time being.

Once the personalization data is saved, the system then allows the user to access personalized recommendations chosen using the personalization keywords. These recommendations can have a number of forces placed upon it, and they are discussed in the personalization section of this document. The recommendation choice passes through an “explanation” module that attempts to translate the reasoning behind the choice into English. For example, if there is a very high degree of matching on a particular topic, the system may use that topic as part of the explanation. On the other hand, if the topic has a broad range of matches, the module uses more general terms or a list of the few top matches to explain the suggestion. However, the system can also generate suggestions that have low direct correlation, and these matches are usually described very generally. Lastly, each possible mode also has several different presentations in terms of sentence structure, wording, and level of detail, which is used to prevent the system from becoming monotonic.

System Design

The system is designed in three distinct parts: the media lab information gathering front end, the database and reasoning backend, and the kiosk front end. All processing is concentrated in the backend, allowing the other parts to function relatively independently of machine specifications. Each subsystem has its own architecture and uses a different language that is chosen to optimally leverage the specific requirements of each subsystem. The data gathering front end is written in PHP on the server side, with HTML and scripting on the client side designed for maximal portability with minimal requirements (only a recent release of a web browser is needed). This choice is based on the requirement that the subsystem be easy to use and adopt. The kiosk front end is written in Java in order to meet the graphical

requirements. Java is also more portable than C, and allows for faster development and debugging, which is important for a deployed system where I cannot be at its side to watch for idiosyncrasies. The final portion, the processing and storage backend, was written in PHP. Though PHP is generally used for web services, it is a capable language that offers excellent database integration and real-time debugging facilities. Additionally, the stateless nature of web-based access allows for easy management of services and “live debugging”, which is impossible with JSP, the other contender for this portion.

The information gathering front end is implemented as a dynamic page that allows authorized users to add/edit the topics of interest to each group and project, and rate the relevance of each topic to the project or group. Un-edited projects “inherit” the information of their group, but with a penalty in the latter processing. This is done because a group generally does not represent its projects (though perhaps the reverse could be considered true). However, it allows projects with no information to still match if no other projects match.

The kiosk front end, described in detail in the interaction design section, is centered on robustness and usability across different machines. It does minimal processing itself, instead farming out the actual work to the main backend server. However, it renders all information and has a large amount of the code (~50%) of the project. It is primarily designed for ease of use, with custom designed widgets intended to give an impression of completeness and finish, which is important to the way in which the system projects itself relationally.

The back end acts as a message router to a number of servers that handle different parts of the badge network. However, primarily of interest from the point of view of the class in the “explanation module”, which handles the processing of internal representation of the choices of the recommender into human-understandable format. The choices are made through several modes of operation. First, there is the direct feature vector match relevancy. Essentially, the topics chosen by the users forms a feature vector with some weight attached to each topic. This vector is multiplied by the matrix of topic choice relevancies from the media lab project database, and the resultant dot products are used as a first pass match relevancy. This is done over the space of all projects and all groups, and the best matches are selected. Groups are penalized as suggestions over project matches, since the group descriptions are usually more “blanket” descriptions than the individual project descriptions, and tend to automatically match more users. If the project selection relevancy is too low (taking into account projects and groups already seen), a group selection is made. The system can optionally also highlight a project or group that has a match in a particular aspect, but is not necessarily highly rated. This prevents to some extent the “more of the same” phenomenon seen with recommender systems. Additionally, in case of failure of all matches, the system has a “random mode” fallback that chooses projects based on popularity of matches against all sponsors.

This should actually have operated by matching against all members within the same company group, but there was insufficient time to load the data and integrate it. An extension was also planned for matching against other sponsors, but this functionality was scrapped due to its parallels with another project from Sandy Pentland's group to be run on the badge as well.

Goals for the Class

The primary goal of the project was to build a relationship with the user very quickly, and then attempt to get the user to return to use the system. User interaction with the system was a high priority from the beginning, since it was unclear in what way the users would react to the system, and how much and at what level they would relate to it. The system is essentially designed to produce an impression somewhere between a watchful and competent proverbial English butler, and a knowledgeable and friendly real estate agent. However, the class readings suggest cases where the expected and the actual outcomes of the interaction were different, and whether this sort of a reaction would take place was unclear. Overall user experience was also an important factor, since obviously it affects repeated use over time. Lastly, I was highly interested in finding out how the users related to the system and how they viewed it, since it is often the case that, with such short experiences, the user receives only an incomplete idea of the relational aspect.

Observations from Deployment

Even though this observation is general, it is important to note that a deployment is not the same as a demo, since there is no one around to fix or explain the situation. Since everything else at the open house is a demo, this is in fact an impediment since people generally outperform machines in relational tasks, especially when they're trying to specifically do so (ie, when trying to draw attention to their projects). Secondly, it is always the case that users are attracted most not to benefits to science, but benefits to themselves. It is important to very clearly portray that benefit for all aspects of the system, in order to gain maximal participation and acceptance. It is also important to note that out-of-context user studies are a dangerous thing. In fact, I must admit to having fallen for this exact flaw. While I did user interviews in an attempt to find out what the sponsors wanted out of the kiosks (and in fact, what everyone wanted out of them), I did not take into account the environment of the demo days. While it is difficult to simulate the conditions, it is equally important to remember that the context will be different and look for hedges against such problems. Failure to do this seriously affected the success of my project, in my opinion, because the timing of each deployment segment and focus of that segment depended too heavily on the original analysis which contained this flaw.

Usage observations

The actual usage of the system can be easily categorized by number of uses. In general, the initial screen was successful in initially hooking people. However, I noticed that the cost of entry was extremely high in the TTT environment, and few people chose to interact further. The location service proved to be the next most popular, since the initial screen promised a direct “this person in here” type response. The exact same maps were used in both cases, in order to meet expectations as suggested by the idea of working relationships. The location service proved to be quite accurate despite what seemed to be a slow update rate. The messenger service was also much easier to use than before, and had a number of users. However, the personalization service had an even higher barrier to entry, and was only used when I demo-ed the service to sponsors. Nonetheless, I was able to receive some feedback from the users about the system through informal interviews. As expected, people found the initial screen compelling enough to investigate the kiosk, but felt too rushed to stop and complete the personalization, which they saw as an “optional activity” that they did not have time for. However, it was unanimously felt that the service to remain available for future use. There was a general expression of the fact that the kiosks were in fact interesting and engaging in terms of usability. However, this engagement was tempered by the barrier to entry, which prevented most individuals from interacting with the system beyond the initial screen. Lastly, the users who used the system believed that the system was trustworthy and capable of explaining itself sufficiently for the user to go where the system suggested they go (or at least investigate the project further). However, this was based on a minimal experience with the system, and the question of whether the system will be trusted in the long run remains very much an open question.

Goal Status and Conclusions

In the theoretical domain, I believe that the design of the system was successful in achieving the relational goals planned for the various components, and I feel that in a less hectic environment, the system may indeed be able to build a better relationship with the user. One of the problems that remain open is the issue of separation of validation of the system’s output in time from the actual even of producing output. Since the user has to leave the system to investigate the recommendation, it becomes problematic to measure compliance and *prima fascia* trust. In general, the users felt connected to the system and felt that the system had a reasonable understanding of their interests, which is a positive result. However, as again the exact interaction between the social scenario of the TTT meet and its impact on the usage of a recommender system that essentially promotes efficiency is unclear.

Next Step

In the next step, a survey will be presented to the sponsors that will attempt to tease out the more general feelings about the kiosk systems. The PLDB annotation services will return to availability for future use by sponsors and media lab members. I would also like to test the exact same software under less harried conditions to try to get a more general idea of how people relate to the system. Moreover, I would like to investigate some of the dynamics of the environment and how that interacts with having a recommender system available in a social environment. In the near future, the system will be available to sponsors via the web for planning visits to the lab. The personalization process and the recommender will also be improved and modified for web access. I will most likely strip out the recommender for the kiosks, and allow access only via web. The kiosk may be used again in the next TTT event, though that is unclear at this time. However, the personalization will be decoupled regardless of their availability.