Collaborative Preference Elicitation

Waterloo Technical Report No. CS-2009-06

Usman Ismail
Cheriton School of Computer Science
University of Waterloo
{uismail}@uwaterloo.ca

1. INTRODUCTION

Early research in the software engineering community discovered the importance of modelling user characteristics and preference in order to make systems more usable [8]. The software of the time tended to be large, monolithic with limited number and type of users performing a limited number of possible tasks. For such systems the community developed concepts such as use-case analysis and structured interviews. Over time sofware has become highly dynamic and scalable. It is not uncommon for modern systems to serve several millions of distinct users spread over many continents. Furthermore systems have become generalized to the extent that the same software (e.g. google search) may be used for tasks as diverse as helping write a school report or shopping for shoes. It is no longer possible to to determine the number, characteristics or skill-set of users when developing the system. In addition social trends such as the prevalence of "web 2.0" has increased users' expectation of usability and personalization.

In such an environment we rely on Preference Elicitation (P.E) to determine user goals, plans, aptitudes, characteristics and preferences. However as much literature has noted [25, 24, 10], this process is non-trivial. Users tend to be irrational, provide conflicting choices and have incomplete knowledge about the domain of the system and thus are unable to provide effective information. Furthermore users dislike being interrupted when performing tasks[7] nor do they like prolonged elicitation sessions[14]. There is a vast array of work that tries to address parts of the preference elicitation problem. This includes work on User Modelling, which seeks to "effectively" elicit user preferences and build a coherent model of the user. In this context "effective" implies, accuracy, completeness and disruption to the users. Once there is a model of the user for the system we use *Constraint Satisfaction* or *Decision Theoretic* approaches to decide which action of the system is most desirable for the user. Lastly we can use the concepts of Voting and Auctions to decide which actions are generally desirable for a group of users with conflicting interests.

Despite the extent of work in Decision Theory and Constraint Satisfaction, we note that many of the algorithms proposed are NP complete for the general case [32]. This necessitates the use of approximations and heuristics. The inherent inaccuracies in the underlying user model compounded with the use of non-deterministic algorithms means that elicitation systems are often unsatisfactory for users.

We address this issue using a technique called collaborative elicitation. We leverage the fact that humans rarely have unique preferences, instead groups of users tend to have similar preferences [28]. This fact has lead to the development and successful use of collaborative filtering techniques. We apply the same concept to preference elicitation, instead of requesting or inferring all preferences from the user we try to categorize users into stereotypes and use default settings for each stereotype. Further we allow the members of each stereotype to determine the defaults for their peers. This is done by allowing user to disable the default setting and specify their own choice. The system uses the specified settings in conjunction with the level expertise and the level of surety of the user to update the default. We note that a single user may not have enough domain knowledge to make an informed decision about all choices nor the time to acquire the domain knowledge. However each user may have or acquire information about a subset of the choices. Hence by allowing users to specify some choices and leave others to their peers we allow the domain knowledge of the whole group to be aggregated.

An important thing to note is that aggregate preference elicitation is not suitable for all situations. For example preference for a color or screen layout is personal and may not benefit significantly from aggregation of group opinions. On the other hand, even in such scenarios it is generally agreed that some color combinations work better together then others. However such aggregated opinions are ideal for a subset of preference elicitation problems that we dub *Configuration problems*. We distinguish a configuration problem from the generic preference elicitation problem in that a preference in a configuration problem has non-trivial consequences for the user. Therefore, unlike a preference for a colour a configuration setting can be right or wrong. However the "right" answer maybe different for different users based on how they use the system. A good example of a configuration problem is a firewall configuration, a liberal firewall policy could allow malware to catastrophically destroy critical user data however being overly conservative reduces user productivity. The exact degree of conservatism to use is highly dependent on the characteristics, goals and cost verses benefit assessment of each user. Such problems are especially suited for aggregate elicitation because it is difficult for a user to follow and understand all new threats to change settings accordingly. However a subset of the users may have the knowledge to understand and react to each of the various threats. We empower those users to make a decision for the whole group. For the remainder of this paper we are targeting the subset of PE problems that can be classified as configuration problems.

In the description above we hinted at several tasks that need to completed before we use aggregate elicitation, these include; categorization of users into stereotypes, assessing their level of expertise and determining how sure they are about each option they specify. There are well known machine learning algorithms available to perform these tasks such as K-Nearest Neighbour [12] and Baysean Classifier [9] however they are have some known problems. Firstly the accuracy of the algorithm is highly dependent on the training sets, however large annotated training sets are not generally available for arbitrary domains. Also learning algorithms suffer from *Concept Creep*, where the categorization of future concepts is guided by the concepts submited earlier. Lastly learning algorithms can be slow to react to changes in concept space. To address this issue we also utilise the concept of user collaboration. We use users to categorize other users using *Games with a Purpose* (See section 2.3). We also use games to determine the skill level and domain knowledge of users (See Section 2.3 for details).

The rest of this paper is organized as follows: Section 2 presents an overview of group preference elicitation, collaborative filtering and games with a purpose. Section 3 discusses the goals and requirements of our system. Following that Section 4 details the design of our proposed system. In Section 5 we present a sample application of our approach to preference elicitation. Finally in Section 6 we present some limitations of our work that will be addressed in future work followed by summary and conclusions in Section 7.

2. BACKGROUND

2.1 Prior Work

The majority of the in preference elicitation of groups of users focuses preferences for a single global outcome [17, 18, 20, 23]. MusicFX[23] selects a playlist for a fitness center based on the preferences of the users present at any given time. Let's Browse[20] provides webpage recommendations for two or more users using the browser collaboratively. Intrigue [18] recommends tourist attractions for a heterogeneous group of users who must travel together. Papers such as Jameson et al[17] even go to the length of building an intelligent agent to represent the user if she is not available for communication. The emphasis of the work is to elicit responses form users independently and then present a global aggregate user-prefernce model. We wish to tackle the more common scenario of a group of users having similar individually distinct user-models.

There has been some preliminary work that seeks to use collaboration between users to help preference elicitation including Plua et al.[26]. The authors propose a group travel recomender system with collaborative elicitation among users. However we note that the authors constrain their design to users who already know each other and are willing to spend time and effort to explicitly verify the preferences of others and to provide suggestions explicitly for specific users. Although this model may be suitable for a family (the group used in the paper) it is not scalable to a larger group of users who may not know each other well enough to provide explicit suggestions nor be willing to expend the effort to verify the preferences of others. Further the work only looks at groups who have a single common goal. This again is a valid assumption for a family going on vacation but breaks down with a generalized group of users.

In general we note that all of these approaches constrain the users to be geographically co-located (for at least part of the time) and to have actual social interaction transcending the system. However the biggest use for elicitation is in distributed applications that run over the Internet. In such a scenario many users have little or no direct social interaction. In fact many users may never interact outside the system for which elicitation is being performed. In many systems the users may not even interact directly at all, they may only be passively influenced by the actions of others. (Virtual community [16]) To the best of our knowledge excluding collaborative filtering there is no work that seeks to perform elicitation for a disjoint group of people with varying goals. We discuss collaborative filtering in detail in section 2.2. However we note that collaborative filtering is not sufficient to tackle the generalized elicitation problem because it can only provide recommendations or rank ordering for a group of elements. It does not help users express their abstract preferences or requirements for the operation of a system.

2.2 Collaborative Filtering

Collaborative Filtering (CF) [16] is a technique for filtering a large set of elements to just those that would of interest to a user based on the opinions of the user's community. Malone et al. [22] define three types of filtering tasks including; cognitive, economic and social filtering. Cognitive filtering uses the content of the elements, economic filtering uses the cost of searching and benefit of use and social filtering uses judgements of quality by a group of users. Social filtering has been very successful in making automated recomender systems and is almost in e-commerce websites.

The users of such a system "rate" a set of common elements based on relevance, preference or perceived quality. People who give "similar" ratings to "similar" elements are clustered together into groups. Such users are said to be members of a *Virtual Community*; they do not directly interact but their actions influence the actions of others as if there is interaction within the group. It is assumed that the goals and payoffs of users in one group are also similar. Using this assumption we can use the aggregate opinion of the group to filter elements for individual users. A major challenge is to define *similar* elements and ratings. There are a number of schemes proposed to calculate similarity including K-Nearest Neighbour algorithm [12] and Bayesian Classification [9].

Although Collaborative filtering has had much commercial success it does have some drawbacks including difficulties with a cold start and the fact that it encourages "Follow the Herd" mentality. A new element with no ratings is unlikely to be brought to users' attention and therefore unlikely to be rated. In addition if a set of people rate an element highly in quick succession then it may receive a higher rating than it deserves or will receive in the long run. Many sites deal with issues by using a parallel expert rating (Editor's rating [1]) system to rank new items. Other websites do not display any ratings until an element has been rated a certain number of times [3]. However a deeper problem yet to be addressed is that ratings are subjective and hence not directly comparable across users. For example some users may be "lenient" scorers and assign higher scores on average than others. Hence a score of 6 from one user may not be equal to the same score from some other user. Also users' perception of the value of elements changes over time or based on their short term goals or moods. This may influence their rating and once the rating is set it is not frequently changed. Lastly a malicious user can unnaturally increase or decrease the rating of an element by using techniques such as Sybil Attacks [13]. In such an attack a single user gets access to multiple accounts and rates an element using all the accounts. A similar attack is to form a cartel of users who rate elements in exchange for monetary reward or reciprocal ratings. In order to address such issues Seth et al. [31] explore the use of social networks to derive a trust model and differentiated ratings based on the level of trust of the person generating the rating. Despite these problems collaborative filtering operates reasonably well in practical systems and in some scenarios even out preforms expert advice [16].

2.2.1 Parallels between CF and PE

The preference elicitation problem can be generalized to two core questions; (1) What? and (2) How? What content to show the user, this may be dynamically generated content (automated schedules [6]) or just a selection from a large body of content (personalized news feeds [5]). The second question is how to present the content, this includes aesthetic considerations, users capabilities as well as characteristics such as language she can understand. Collaborative filtering also tries to address the "'What?' question by using the hypothesis that similar users should be shown similar content. In fact both PS and CF can be placed in Malone et al.'s taxonomy of content filtering methods, PE falls under the cognitive filtering heading where as collaborative filtering falls under the social filtering category. Therefore the goals of a collaborative filtering approach are a subset of the goals of preference elicitation.

An example of preference elicitation systems that are very similar to collaborative filtering is Adaptive News filtering systems such as YourNews[4]. The system crawls articles from a number of news feeds and builds an abstract representation of each article in the form of a Term-Vector [29]. There is also a similar representation of the users "interest" in the form of keywords with weights representing relative importance. Using cosine distance the system computes the relevance of an article to the user and presents articles that or greater relevance. The GroupLense [19] project provides a similar service using collaborative filtering.

In addition to common goals CF and PE also face similar challenges and employ many similar techniques to address those challenges. Both techniques need to "know" their user. In CF we need to classify the users and cluster them into groups of similar users whose actions will be used to guide the actions of others in the group. In PE we need to infer the users' reaction to an item by reasoning based on their goals, characteristics and preferences. Therefore both CF and PE use similar user modelling approaches such as Stereotypes[28] and Ontologies [27, 30, 11]. In building such models of individual users systems face similar challenges of incomplete and inconsistent information about users. Nisbett et al.[25] note that people are notoriously unreliable sources of information about themselves. In fact McGuire et al. [24] find that people describe themselves based on the context they perceive themselves to be in. Therefore both PE and CF try to use indirect means to elicit information about users. CF systems have a definite advantage in this area because they use aggregate group preferences as opposed to individual ones. Hence inconsistencies based on a particular user's input are obfuscated. For example in a Movie recomender system a user may rate a movie poorly because she had a bad experience in the theatre (movie was delayed or noisy audience). If PE was used to build the system then the system may rank movies from the same genre, director or studio poorly. A collaborative filtering system will be less likely to suffer similar problems because it aggregates opinions of people from different locations who saw the movie at different times in different theaters.

2.3 Games With a Purpose

There have been huge advances in computing hardware and software over the past several decades, however some tasks remain fundamentally "hard" for computers but are trivial for most humans. Tasks requiring conceptual intelligence and perceptive processing are especially hard for computers. This has lead researchers including Von Ahn et al.

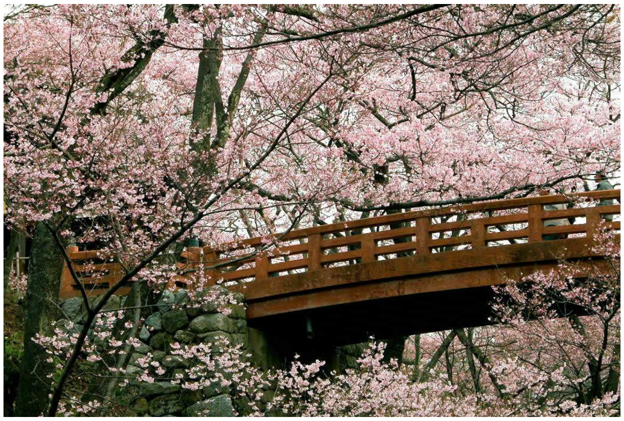

Figure 1: ESP Sample Image

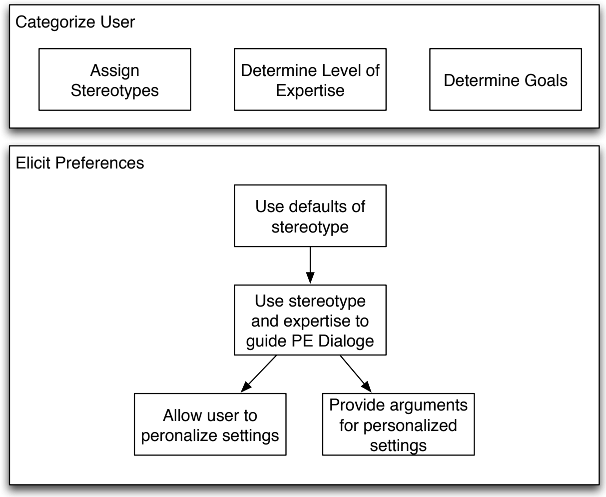

Figure 2: System Design Overview

[33] study the idea of using humans as processing agents for computer systems. Von Ahn notes that people spend millions of hours playing games which require the same inference, pattern recognition and reasoning skills that are required by many modern computer systems. If peoples processing capacity could be channeled towards constructive purposes many very complex computation problems could be solved. Unlike computers people need incentives to perform work and therefore Von Ahn et al. developed the concept of Games With a Purpose. The game provides an environment for users to compete with each other and perform a tasks that are considered "fun" thus providing incentives to participate. However as a side effect of the game play the system is able to derive useful information from the users actions. Such games are especially suited to tasks such as relating concept and categorization of elements. Two games that highlight these aspects respectively are Verbosity [34] and ESP[2].

Verbosity is a game for collecting relations between concepts that are not obvious through semantic rules. For example in one instance of the game the word "Axe" elicited related concepts of "Cutting trees", "Wood Chopping". The game is played between two players, one is shown a word (Axe) and she has to enter phrases that describe the concept to help her partner guess the word. However the first player may not use words or phrases that are directly related to the secret word. This prompts users to enter phrases that are non-trivially related to the secret word. We also have explicit feedback on how strong the relation is because we know if it resulted in a correct guess.

The ESP game involves the same setup of a pair of collaborative players but the players are both shown an image. Both players use words or phrases to describe the image. If both players enter a common phrase then they are both awarded points. This generates annotations for each images that can be used for indexing purposes. While some of the labels could have been generated by advanced image processing algorithms the game generates much more rich annotations. For example Figure 1 shows an image that may generate the tags "Bridge" or "Tree" If it is precessed in an computer vision system but the game generated tags such as Cherry Blossom, Park, Forrest, Spring.

The concepts and annotations generated by such games can be used directly to relate elements but additionally the can also be used as training sets for machine learning algorithms. In section 2.3 we specify several games that are used by our system to help categorize users and into stereotypes as well as annotate users and settings to help elicitation.

3. GOALS OF THE SYSTEM

We propose to build a framework that allows users to perform a complex configuration task with minimum required effort without compromising correctness.This can be divided into several separate sub-goals; (1) accurate categorization of users into groups (Stereotypes), (2) generating acceptable default settings for each groups, (3) determining the level of skill and aptitude of users, (4) allowing users to learn about settings which are of greatest relevance to them and (5) making informed decisions about selections about which they have domain knowledge. Section 4 provides details about how each of the goals is achieved by our system.

4. SYSTEM DESIGN

Figure 2 shows an overview of our system. There are two phases of system operation, first we collect information about the user and assign a stereotype to each user (See Section 4.1). We also assess the user's skill level and aptitude and determine common tasks (or goals) that the user might perform as discussed in section 4.3 and section 4.2 respectively. Once we have finished assessing the user we use the information to guide the elicitation process in the second

Section	Sub-Goal Addressed
Assigning Sterotypes (Section4.1)	1
Assigning Goals (Section4.2)	2
Determinging Skill (Section4.3)	2,3
Determining Defaults (Section4.4)	2
Guideing Eclisitation (Section4.5)	4,5
Generating Meta Data (Section4.6)	4

Table 1: The sub-goals addressed by each section

phase. Using the stereotypes of each user we determine default values for all settings for the user (See section 4.4 for details). We also use the goals to determine which settings are of particular relevance to the user and prompt the user to find out more about those settings. The system provides, user generated, information about the setting and arguments for and against choosing each of the possible options. We also ask the user to specify the level of confidence for each of those settings. We next describe each of the systems components in greater detail. Finally Table 1 lists how each of the sub-goals identified in section 3 are addressed by one or more of the componenets of our proposed system.

4.1 User Categorization

The first stage in our approach is to categorize users into stereotypes, to achieve this we allow the user to describe themselves in free text. We use free text descriptions because they provide rich high quality information about elements Goldberg et al. [14]. We then assign keywords or tags to the user by using a game similar to ESP [2]. and show the description to two "players"who have to assign a word (e.g. Student) or phrase (e.g. marketing executive) to describe the user. If the two players select the same word or phrase then they both score a point and move to the next description. The two players have unlimited guesses but a limited amount of time to make guesses. Guesses that frequently lead to a point being scored are marked as "good" keyword to describe the user. To generate a diverse set of tags words and phrases that are already ranked very highly are disallowed. To provide incentive for game players the highest ranked players at the end of each month can be given a reward. Each of the keywords that is marked as "good" will be assigned a stereotype to the author of the description.

4.2 Assigning Goals

We now have stereotypes for each user but those stereotypes have no meaning as we have no prior knowledge about what each stereotype implies. In the context of a configuration task a goal translates into an action that the user may perform, the configuration parameters that relevant to that action. For example if our task is to configure a network firewall then a user assigned a stereotype "IT Specialist" may need to run network monitoring applications such as trace-route band needs the firewall to allow such applications network access. However a user assigned a stereotype "Marketing Executive" is unlikely to need such applications and allowing such applications may expose the user to unnecessary security vulnerabilities. To assign goals to each stereotype we again build a game. We leverage the fact that a configuration task has a finite number of parameters. We ask each user to describe the tasks they use the system for or envisage themselves performing using the system. We show the description to a pair of players and ask them to select a configuration parameter that is of relevance to the user. If both players select the same parameter they score a point. As before players have unlimited guesses but limited time to make the guesses. We mark all the parameters commonly selected for a description as important parameters for the author of the description. In addition we also mark those parameters as important for all stereotypes assigned to the author of the description.

4.3 Determining User Aptitude

To determine user aptitude we propose the use of a quiz relating to the system, each right answer awards the user one point and each incorrect answer deducts a points from the user. Users will be asked questions relating to the configuration parameters identified as important for their stereotype using the procedure in section 4.2. This encourages users to find out about parameters that are of relevance to them and make informed decisions about those parameters. In addition the score will generate a rank ordering among users belonging to each stereotype which allows us to judge how much trust to put in their judgment.

4.4 Determining Defaults

We now have enough information begin assigning default values to the various settings. For each setting we determine which stereotypes mark the setting as important. For all stereotypes from the set that marked the settings as important we select the ones that are assigned to the user. All users that belong to any of these stereotypes are considered "peers" for the setting in question. The settings selected by the peers are used to determine the default value of the setting using weighted plurality voting [21] similar to approaches used in collaborative filtering. If a peer selected "Use Default" for a setting then their vote is omitted. If the user selected a value for the setting then their vote is weighted to reflect their skill level, as determined using the procedure in section 4.3 and their level of confidence in their choice. The confidence of the user is ascertained by asking the user the following question, "What percentage of your peers should be in consensus before your selection is overruled?". This provides the user with incentive to tell the trust about her level of confidence in the selection. By lying and reporting a larger value she risks being vulnerable to environmental factors that make her choice unsuitable. Returning to the firewall example if there is a new threat which prompts many users to switch to a more conservative firewall setting. By reporting a large value of confidence the user prevents the system from reacting to the shift in user opinion and adjusting the firewall settings. The option that wins the weighted plurality vote for each parameter is selected as the default option.

4.5 Guiding elicitation using stereotypes

The users are prompted to make their own selections for those parameters that are marked as important for one or more of their stereotypes. Users are able to adjust other parameters but they will not be prompted by the system to do so. When the system prompts a user to specify a setting, it provides the user an option to "Use default" which uses the aggregate as calculated in section 4.4. With each option the system provides a link to a description of the setting, and with each possible option the system provides arguments for or against selecting that option. These descriptions and arguments which we dub *Meta-Information* are generated by the users themselves using the procedure described in section 4.6.

4.6 Generating Meta-Information to help Preference Elicitation

One of the important challenges in preference elicitation is prosed by users with incomplete domain information. The best way to tackle this problem is to provide information about the implications of preferences and the various possible choices that the user can make. However generating descriptions for all parameters and choices for those parameters targeted at users of different backgrounds and skill levels is difficult and costly. We propose the use of collaborative approaches to generating meta data about the various options. Firstly we develop a game to generate descriptions of the various parameters. A pair of players are selected, one player is shown an options and must make her partner guess the option without using its name or related phrases. If her partner is able to guess the option being described then the descriptions is marked as a good description. In order to generate repeated ratings for the same description players have the option of viewing a random previous description for partial points. This will generate many descriptions of the parameter which can be ordered in terms of "quality". A user is shown descriptions given by other users belonging to her stereotype, this increase the relevancy of the description for the user.

In order to generate arguments for or against a particular choice of a particular option we ask all users to give a reason why they selected the choice which they did. In addition users who read the arguments can rate the argument by answering the question, "Did this argument convince you to make a decision?" If the answer is yes then the argument is marked as a "good" argument. Such a system of rating feedback is successfully used by many collaborative filtering systems including IMDB [3]. To add incentives for a user to give a "good" argument we can add periodic rewards for arguments that were successful in convincing the largest number of users.

Figure 3: Prompt user to find out more

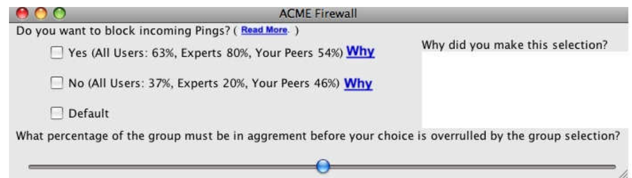

Figure 4: Sample interface

5. APPLICATION

We have already used the example of a firewall configuration several times in this paper because it is an ideal configuration task with regard to collaborative elicitation. Configuring a firewall is a complex task but it is important to get the configuration right because a configured firewall can lead to security holes or prevent users from performing legitimate tasks. We now describe a collaborative firewall configuration agent in detail.

After installation the firewall parameters will be set to an initial default, the defaults are usually the most conservative settings commonly used by users. This is the way most commercial firewalls behave. However the firewall will also ask the user to submit a free text description herself as well as the common network related operations she performs and the applications she commonly uses. This description is updated in the on-line database of the Categorize user and Assigning Goals games. Once the games generate set of tags and goals for the user the configuration agent will determine the user's peers and important settings and use aggregate opinions to adjust the defaults. The firewall will also prompt the user to find out more about configuration properties that may be of relevance to her as they are invoked. For example Figure 3 shows a message that the firewall may display if it blocks an application from running because of a default settings to do so.

If the user elects to change a setting based on her own initiative or prompting from the software she will be shown an interface similar to the one in Figure 4. It will provide links to find out about the setting, and arguments for selecting each possible choice. It will also allow her to specify why she made her choice and the level of confidence she has in her choice.

After installing the firewall the user is also asked to log into a web interface where she can participate in the games and answer questions on the skill determination quiz. The interface will show her skill level (initially beginner or novice), her score in the various games and ranking in relation to other players. Players should also be able to communicate as this promotes a sense of community and will encourage contributions to the system via game playing and meta-data generation.

6. LIMITATIONS AND FUTURE WORK

Although our approach presents many opportunity for improvements to the current state of the art in elicitation there are some limitations or our approach that are left to future work. Firstly although free text descriptions have many advantages there are some drawbacks. Firstly it adds additional burden on the user which was shown to be problematic in [14]. Furthermore when free text is used for descriptions of why users made a particular choice we come to Grudin's[15] question of "Who does the work and who gets the benefit?" However we note that unlike [14] our system does not make entry of free text descriptions a recurring task. Instead the user will infrequently change firewall settings and generate associated descriptions. To answer Grudin's question we provide incentives to entering descriptions by rewarding descriptions that are rates highly by users. In addition a casual survey of any of the collaborative filtering systems such as IMDB [3] shows that a large number of users are willing to generate textual descriptions with little or no incentive. We propose to study the willingness of users to generate free text descriptions in our system to evaluate this approach.

Another drawback of our approach stems from the fact that we use human computation and collaborative tasks extensively. Such a system needs a large "critical mass" of users before it is able to operate effectively. However if the initial results of the system are not pro missing then the user-base is unlikely to grow to the critical mass. An ideal situation would be to apply these concepts to a system that is already in operation and thus start with a large user base and backup system while meta-data and tags are being collected. We plan to conduct a study of the number of users required before the system is able to sustain itself.

The task of finding peer groups for settings and aggregating opinions of the peer groups can be very computationally expensive if the user-base grows large. This will pose a scalability challenge for the system. However we note that most of the computationally expensive tasks can be performed offline and large data centers are able to provide more then enough processing capacity to handle several millions of users. We plan to implement the system and run it using test loads to determine the computational requirements.

Lastly by using collaborative elicitation we may have added opportunities for manipulation by malicious users. This can be in the form of a Sybil Attack [13] where a single user gains control over multiple accounts and influences group decisions. In addition user-generated content may contain incorrect and misleading information. We have not as yet explored the the possible attacks by malicious users and possible security measures to prevent them. We plan to explore these issues in future work.

7. CONCLUSIONS

There is great scope of advancing the field of preference elicitation by using the the power of human computation. Users already generate vast amounts of support content on the web with little or no direct incentive to do so. By channeling this instinct for contributing to the *common goad* and adding some incentive structures we can tackle some of the most challenging aspects in preference elicitation and artificial intelligence by using real intelligence. Although this work looks at directly using human computation to achieve our goals the human computations can serve to complement artificial intelligence algorithms and machine learning techniques. We feel this model of using large scale human computation is an ideal tool to help solve preference elicitation problems and hope that it will be studied in the context of many such systems.

8. REFERENCES

[1] Cnet: Download.com, www.download.com, November 2008.

[2] Games with a purpose: The esp game, November 2008.

[3] Internet movie database, www.imdb.com, November 2008.

[4] Jae-Wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Y. Syn. Open user profiles for adaptive news systems: help or harm? In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 11–20, New York, NY, USA, 2007. ACM Press.

[5] Liliana Ardissono, Luca Console, and Ilaria Torre. An adaptive system for the personalized access.

[6] Alexander Babanov, John Collins, and Maria Gini. Scheduling tasks with precedence constraints to solicit desirable bid combinations. In *In Proc. of the Second Intl Conf. on Autonomous Agents and Multi-Agent Systems*, pages 345–352, 2003.

[7] Brian P. Bailey, Joseph A. Konstan, and John V. Carlis. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings INTERACT 01*, pages 593–601. IOS Press, 2001.

[8] David Benyon. Adaptive systems: A solution to usability problems. *Journal of User Modelling and User-Adapted Interaction, Kluwer*, 3:1–22, 1993.

[9] Scott Buffett and Bruce Spencer. A bayesian classifier for learning opponents' preferences in multi-object

automated negotiation. *Electron. Commer. Rec. Appl.*, 6(3):274–284, 2007.

[10] Robert Bushey, Jennifer Mitchel Mauney, and Thomas Deelman. The development of behavior-based user models for a computer system. In *UM '99: Proceedings of the seventh international conference on User modeling*, pages 109–118, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[11] Gil Chamiel and Maurice Pagnucco. Exploiting ontological information for reasoning with preferences. In *Multidisciplinary Workshop on Advances in Preference Handling (MPREF'08). In conjunction with AAAI2008*, 2008.

[12] Thierry Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25:804–813, 1995.

[13] John R. Douceur and Judith S. Donath. The sybil attack. In *First International Workshop on Peer-to-Peer Systems*, pages 251–260, 2002.

[14] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[15] Jonathan Grudin. Social evaluation of the user interface: Who does the work and who gets the benefit. *Proceedings of IFIP INTERACT'87*, pages 805–811, 1987.

[16] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[17] Anthony Jameson, Stephan Baldes, and Thomas Kleinbauer. Generative models of group members as support for group collaboration. In *Proceedings of the UM 2003 Workshop on User and Group Models for Web-Based Adaptive*, 2003.

[18] Anthony Jameson, Stephan Baldes, and Thomas Kleinbauer. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. In *Applied Artifcial Intelligence*, 2003.

[19] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.

[20] Henry Lieberman, Neil Van Dyke, and Adriana Vivacqua. Lets browse: A collaborative web browsing agent. In *In Proc. Intl. Conf. on Intelligent User Interfaces*, pages 65–68. ACM Press, 1999.

[21] W.F. Lucas. Measuring power in weighted voting systems. *Case Studies in Applied Mathematics*, 1976.

[22] Thomas W Malone, Kenneth R Grant, Franklyn A Turbak, Stephen A Brobst, and Michael D Cohen. Intelligent information-sharing systems. *Commun. ACM*, 30(5):390–402, 1987.

[23] Joseph F. Mccarthy and Theodore D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372, New York, NY, USA, 1998. ACM Press.

[24] William J. McGuire and Alice Padawer-Singer. Trait salience in the spontaneous self-concept. In *Journal of Personality and Social Psychology*, volume 33(6), 1976.

[25] Richard E. Nisbett and Timothy D. Wilson. Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84(3):231–259, March 1977.

[26] Claudia Plua and Anthony Jameson. Collaborative preference elicitation in a group travel recommender system. In *Proceedings of the AH 2002 Workshop on Recommendation and Personalization in eCommerce*, pages 148–154, 2002.

[27] Liana Razmerita, Albert Angehrn, and Alexander Maedche. Ontology-based user modeling for knowledge management systems. page 148. 2003.

[28] E. Rich. User modeling via stereotypes. In *Cognitive Science: A Multidisciplinary Journal*, volume Vol. 3, No. 4, pages 329–354, 1979.

[29] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[30] Vincent Schickel-Zuber and Boi Faltings. Inferring User's Preferences using Ontologies. In *AAAI 2006*, pages 1413–1418, 2006.

[31] Aaditeshwar Seth, Jie Zhang, and Robin Cohen. A subjective credibility model for participatory media. *AAAI Workshop on Recommender Systems*, 2008.

[32] Larry Joseph Stockmeyer. Measuring power in weighted voting systems. *Massachusetts Institute of Technology. Dept. of Electrical Engineering. Thesis.*, 1974.

[33] L. von Ahn. Games with a purpose. *Computer*, 39(6):92–94, June 2006.

[34] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78, New York, NY, USA, 2006. ACM.