# What a Difference a Group Makes: Web-Based Recommendations for Interrelated Users

Anthony Jameson[1] and Barry Smyth[2]

[1] DFKI, German Research Center for Artificial Intelligence
[2] Department of Computer Science, University College Dublin

**Abstract.** Whereas the other chapters in this book deal mainly with adaptation to individual users, web-based personalization must in fact often be directed at groups of users. The shift of focus from an individual to a group makes more of a difference than one might expect at first glance. This chapter looks at the new issues that arise when one considers web-based personalization that involves groups. For concreteness, we focus on the subclass of group recommender systems; but we also point to implications for other types of personalization for groups. The chapter is organized around four general issues that arise with group recommender systems, which concern the specification of preferences, the aggregation of preferences, the explanation of recommendations, and the support of final decision making, respectively. We discuss ways in which these issues have been dealt with in a representative set of group recommenders. To illustrate how these analyses can guide and stimulate future work in this area, we also suggest some ways in which existing systems might be extended.

## 1 Introduction

Almost all of the techniques of web-based personalization discussed in the other chapters of this book are designed to allow effective adaptation to individual users. But often the users of such systems operate not individually but in groups, which may vary from formally established, long-term groups to ad hoc collections of individuals who use a system together on a particular occasion. This phenomenon can in principle occur with just about any form of web personalization. In this chapter, we will focus on the subclass of recommender systems (cf. the chapters in this volume by Pazzani, by Smyth, by Burke, and by Goy et al.), but many of the points made will be applicable by analogy to other types of adaptive web-based system (cf. Section 6.2).

Many common recommendation scenarios are intimately connected with well-defined groups of individuals: friends collaborating with a recommender system to design the perfect vacation; a family selecting a movie or TV show to watch together; a group of colleagues choosing a restaurant for an evening out. In each of these *group recommendation* scenarios, the recommender system must consider explicitly the need to aggregate the diverse preferences of individual users in order to make a recommendation that to an adequate extent satisfies each individual and the group as a whole.

In this chapter, we will identify the issues that must be addressed by group recommender systems and the ways in which they have been dealt with in systems that have

Table 1. Overview of the issues to be addressed in this chapter, organized in terms of the four main phases of the group recommendation process.

| | Phase of the recommendation process | Difference from recommendation to individuals | General issue raised |
|---|---|---|---|
| 1. | Members specify their preferences. | It may be desirable for members to examine each other's preference specifications. | What benefits and drawbacks can such examination have, and how can it be supported by the system? |
| 2. | The system generates recommendations. | Some procedure for aggregating preferences must be applied. | What conditions to such aggregation procedures have to fulfill; and what kinds of procedure tend to fulfill them? |
| 3. | The system presents recommendations to the members. | The (possibly different) suitability of a solution for the individual members becomes an important aspect of a solution. | How can relevant information about suitability for individual members be presented effectively? |
| 4. | Members decide which recommendation (if any) to accept. | The final decision is not necessarily made by a single person; negotiation may be required. | How can the system support the process of arriving at a final decision, in particular when members cannot engage in face–to–face discussion? |

been developed so far. In doing so we will draw on a number of existing group recommender systems to serve as explanatory case studies in relation to certain of these issues.

## 1.1 Overview of Novel Issues

Relative to recommendation for individuals, there are several new issues that arise with the concept of group recommendation. Table 1 gives an overview of the issues to be addressed in this chapter. On the face, of it these are standard issues that are present in all recommender systems, whether group-based or not, but in reality, as we will see, there are important factors that must be considered in group recommendation scenarios that simply do not exist in other forms of recommendation. For example, in traditional (single-user) recommender systems, preference specification is generally something that is performed by each user on their own. In many group recommendation scenarios, however, it can make sense to allow group members to examine each other's preferences in order to facilitate a more effective form of preference elicitation (see Section 2). Separately, the manner in which recommendations may be generated for a group of users, as opposed to the lone user, needs to take a number of unique factors into consideration. One is the issue of how to select items for recommendation that will satisfy the group as a whole, where the concept of "satisfying the group as a whole" is itself a complex notion. As we will see, one possible approach here is that of aggregating the preferences of individuals to provide a representation of group preferences. Conventional recommender systems do not generally concern themselves with such issues, although some related issues are dealt with. For instance, collaborative filtering systems often attempt to model explicit clusters of similar users that reflect groups of users with common preferences. Still, we argue that this type of approach can be distinguished from group recommendation for a number of reasons. First and foremost, to the extent to which collaborative filtering systems attempt to model groups of users, they do so on the basis of user similarity, bringing together users because of shared preferences and rating patterns. In contrast, group recommender systems cannot rely on such homogeneous collections of users. Generally speaking, a social group of users will share some preferences but differ in others. They are a group because of their social context and not because of their rating history. As a result, preference aggregation is a far more challenging issue. Likewise, the generation of a set of items for recommendation is made all the more difficult by the preference conflicts that are likely to exist within a group. How might these conflicts be resolved? Should items be selected because they are moderately acceptable to all group members or should the preferences of certain members be emphasized at the expense of others (see Section 3). Then there is the issue of how best to present recommendations to the group. In particular, there is the need to explain the suitability of recommendations for certain individuals with a view to convincing these individuals that the recommendations represent a good compromise. And finally, the issue of acceptance is very different in group recommendation scenarios than in more traditional recommender systems. For the latter, whether or not a particular recommendation is accepted is a matter for an individual user to decide. With group recommenders, on the other hand, the group as a whole must come to this decision and the issue then becomes how best the recommender system can support this process.

## 1.2 Existing Group Recommenders

We will use as examples six of the group recommenders that have been presented so far in the literature. Since our focus is on general issues rather than on specific systems, for each of these systems we will mention only the aspects that are relevant to the issues under discussion; for further information on the individual systems, please see the references given.

LET'S BROWSE (Lieberman et al. [1]) recommends web pages to a group of two or more persons who are browsing the web together. It estimates the interests of individual users by analyzing their web homepages.

POLYLENS (O'Connor et al. [2]) is a generalization of the MOVIELENS system (cf. the chapter in this volume by Herlocker et al.) that recommends movies to groups of users. The system has more recently been modified to yield BUDDY SEARCH, which makes it easier to form ephemeral groups (see http://www.movielens.umn.edu/).

INTRIGUE (Ardissono et al. [3]) is designed to help tour guides who need to design tours for heterogeneous groups of tourists that include relatively homogeneous subgroups (e.g., "children"). Although it directs its recommendations only at the tour guide, it takes the interests of the various group members into account.

MUSICFX (McCarthy and Anagnost [4]) automatically selects music channels for the music to be played in a fitness studio. On the basis of the preferences that have been previously specified by the members who are currently working out, the system chooses one of 91 possible music channels, including some randomness in the choice procedure in order to ensure variety. Although MUSICFX is not a web-based system, we discuss it in this chapter because it vividly illustrates a number of points that are equally valid for adaptive web-based recommender systems.

The TRAVEL DECISION FORUM (Jameson [5]; Jameson et al. [6]) helps a group of users to agree on the desired attributes of a vacation that they are planning to take together. Special attention is given to support for users who are not collocated and who can therefore not engage in face-to-face discussions.

I-SPY (see, e.g., Smyth et al. [7]) is a community-based web search engine that personalizes search results for a community of like-minded searchers on the basis of a model of community search preferences. I-SPY is a meta-search engine that relies on some underlying search engines to perform the actual query-based search, post-processing the results returned with a view to adapting the ranking of the final result list according to the preferences of a given community of searchers. In particular, I-SPY records the query patterns and selection habits of a community of searchers and uses this information to *promote* (i.e., present earlier in the search result list) results that are favored by the community. I-SPY does not generate its recommendations for a group of searchers who are searching together. It is included in this chapter as an example of a recommender system that focuses on the specification and aggregation of group or community preferences. In turn, I-SPY offers several novel features concerning the way in which its recommendations are presented that should yield ideas for designers of group recommender systems of various types.

In each of the following sections, we will discuss the ways in which one of these issues has been dealt with in previous group recommender systems. The purpose of these discussions is to stimulate thinking about how the corresponding problems can be solved in existing or yet-to-be-designed group recommenders. To illustrate the generative potential of our framework, we will suggest some possible improvements to existing systems.

# 2 Facilitating the Specification of Preferences

## 2.1 Methods for Acquiring Preferences

As we have seen in the chapters by Pazzani, by Smyth, by Burke, and by Goy et al., many recommender systems do not require their users to specify their preferences explicitly. With group recommenders as well, it may be possible for the system to get by with implicitly acquired information about users. For example, LET'S BROWSE estimates the interests of its users by initially analyzing the words that occur in each user's web homepage and then, during group browsing, by analyzing the words that occur in the pages visited by the group. In the case of such implicit acquisition of information about users, there may not be a big difference between the cases of recommendation for individuals and recommendation for groups.

But there are some types of recommender that do require an explicit specification of preferences. For example, MUSICFX needs to know for each visitor to the gym the extent to which he or she likes each of 91 music genres; it would be difficult to acquire

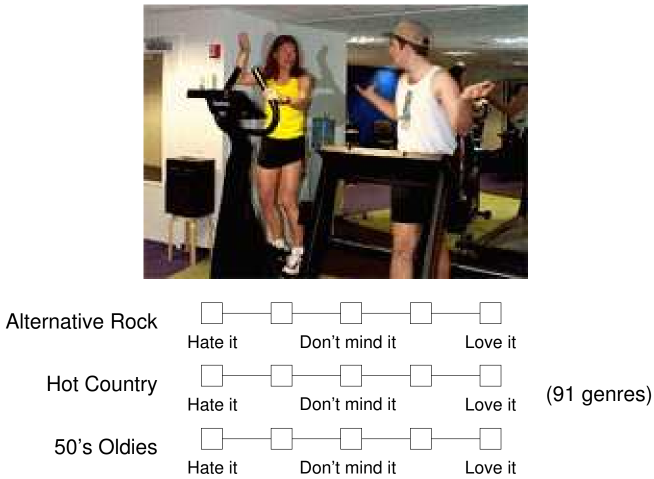



Fig. 1. Preference specification scales for 3 of the 91 music genres covered by MUSICFX.
*(Image from the video of McCarthy and Anagnost [8].)*

such information without directly querying the visitors. Accordingly, each visitor is asked to rate each of the 91 genres on a five-point scale (cf. Figure 1).[1]

Similarly, the TRAVEL DECISION FORUM needs to know how each user feels about dozens of attributes of vacation destinations, ranging from the facilities that are available in their rooms to the sightseeing attractions that are available in the surrounding area. Here again, only explicit elicitation is likely to be feasible.

MOVIELENS, as a collaborative filtering system (cf. the chapter in this volume by Herlocker et al.), falls between these two extremes: Users do not explicitly describe their movie preferences, but they do rate individual movies on a scale from 1 to 5 stars. As we will see, this procedure raises some of the same issues as explicit preference specification.

In I-SPY, the specification of preferences represents an intermediate case between implicit and explicit specification. A user indicates an interest in a given search result by selecting the result in question from a query result list, and I-SPY interprets each result selection as an indication of relevance with respect to the current search query. The specification is implicit in that the user's primary intent in selecting a result is not in general to indicate his or her preferences to the system; but it has some elements of explicitness in that users are aware of the fact that their selections are being interpreted as reflecting their preferences and can, if they like, choose results that they would not otherwise have chosen, in order to influence the system's preference model (cf. Section 3.3).

[1] *Note to reviewers:* Once the selection of figures from previous publications has been finalized, with consideration of the reviewers' recommendations, permission for reproduction will be requested from the copyright holders.

## 2.2 Sharing Information About Specified Preferences

In a recommender system for individuals, there is in general no person besides the user who has an immediate interest in seeing explicitly specified preferences with a view to improving the current recommendation process. In a group recommender, each member may have some interest in knowing the other members' preferences, for several possible reasons:

1. *Saving of effort.* Specifying preferences is usually seen by users as a tedious process. If a group member $m_1$ knows that another member $m_2$ with generally similar preferences has already specified their preferences, $m_1$ may be able to save time and effort by copying at least some of $m_2$'s entries and then perhaps making some changes—especially if the system makes it easy to do such copying and postediting.
2. *Learning from other members.* Another member's preferences may be based in part on knowledge or experience that the current member lacks. For example, if a MUSICFX user notices that his friend has expressed a strong preference for Hawaiian music, he may decide to give it a high rating so as to have a greater chance of being able to give it a try himself.

An attempt to exploit both of these potential benefits is found in the TRAVEL DECISION FORUM: A simple extension of a typical rating-scale dialog box allows the current member optionally to view (and perhaps copy) the preferences already specified by other members (see Figure 2). An additional feature that makes sense mainly if other persons will be viewing the specifications is the option to add brief verbal explanations or *arguments* for specific ratings.[2] Arguments can have various forms and functions in group decision contexts (cf., e.g., Jennings et al. [9]). In a group recommendation context, two typical functions are (a) to persuade other members to specify a similar preference, perhaps by giving them information that they previously lacked; and (b) to explain and justify a member's preference even if the argument is not generalizable to other members (e.g., "I can't go hiking, because of an injury").

Experience with this method of *collaborative preference specification* has revealed further benefits beyond the two already mentioned:

1. *Taking into account attitudes and anticipated behavior of other members*. Sometimes the preference of the current member depends in part on the preferences and/or the anticipated behavior of one or more other members. For example, if $m_1$ sees that $m_2$ has specified a strong preference for tennis facilities, $m_1$ may want to specify a similar preference, reasoning that if a hotel is found that offers tennis, $m_1$ and $m_2$ will be able to play together. Otherwise, $m_1$ may genuinely not want to emphasize tennis facilities, on the grounds that she would probably have no one to play with anyway.
2. *Encouraging assimilation to facilitate the reaching of agreement.* A different reason why $m_1$ may assimilate her preferences to those of $m_2$ is simply a desire to minimize conflicts that may make it more difficult for the group to find a solution. This pattern is especially likely in cases where $m_1$ was originally more or less indifferent between two possible preference specifications, before seeing

[2] These arguments can be entered and viewed in pop-up windows that are not visible in Figure 2.

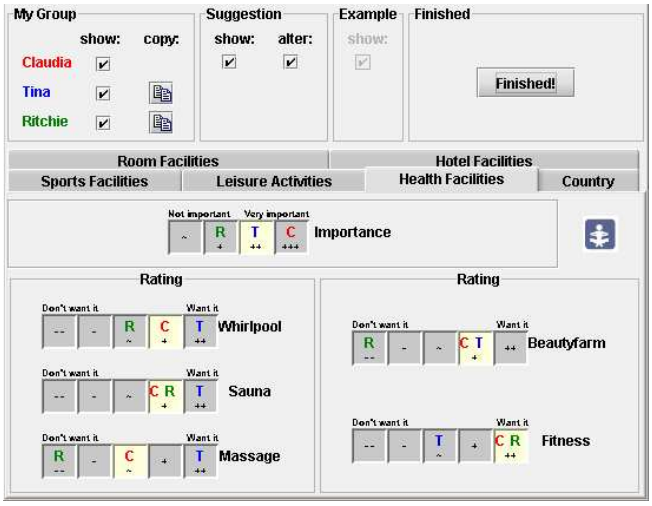



Fig. 2. Dialog box for the collaborative specification of preferences in the TRAVEL DECISION FORUM.

*(The currently active group member is Claudia, the other two are Ritchie and Tina. The preferences of each member are represented by the first letter of their name. Each scale refers to a single attribute and ranges from* $--$ *for "Don't want it" to* $++$ *for "Want it". The highlighting of one cell for each attribute is added only when a compromise proposal has been suggested, as is explained in Section 3.1.)*

that $m_2$ has chosen the other one of them. The difference between this case and the previous one is that here, $m_1$'s true preference has not changed, but she has strategically changed her specification of it.

Although I-SPY's preference specification is largely implicit, there are some phenomena involved in the use of I-SPY that are similar to those that arise with collaborative preference specification of the type we have seen with the TRAVEL DECISION FORUM. These are related to the fact that each user sees the effects of the choices made by other users, even if he does not recognize these effects as such. I-SPY exposes the learned preferences of its community to searchers, in part by highlighting *promoted* results in a search result list (see Figure 3 for examples and Smyth et al. [10], and Smyth et al. [7] for further details). Thus, just as in the TRAVEL DECISION FORUM a user can intentionally copy the preferences specified by another group member, in I-SPY, the choices (and thus the implicit preference specifications) of each community member will tend to be affected by the choices of previous searchers. In recent versions of I-SPY, the current user can even see which individual users were responsible for the promotion of a particular link (see Section 4.2). The purpose of the provision of this sort of information is to provide a sort of explanation of the recommendation of a given link; this function will be discussed in more detail in Section 4; but this property is relevant here in that it can lead to (a) a user thinking twice about whether to choose a particular link because of knowing that others will see that he or she chose that link;

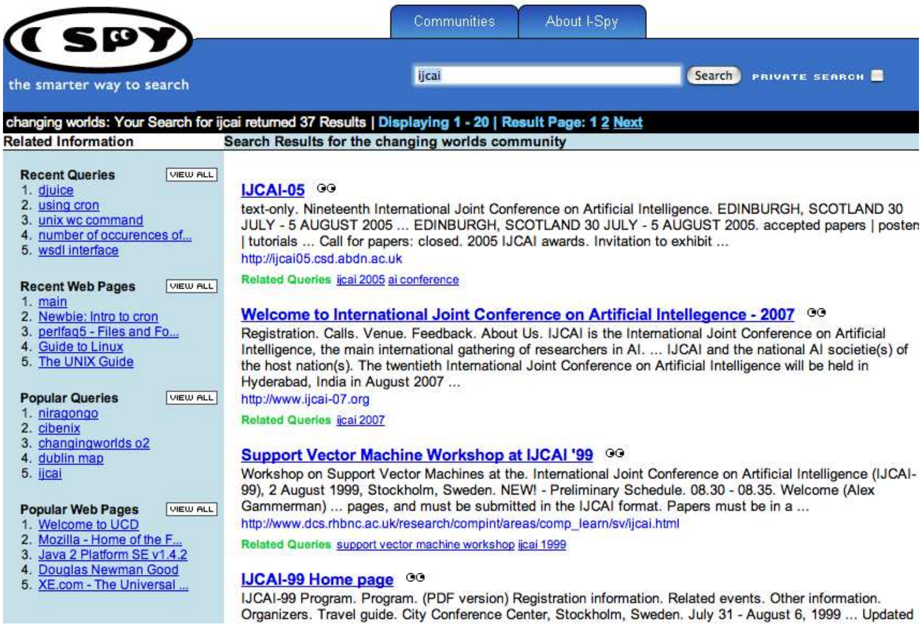



Fig. 3. Screen shot of I-SPY illustrating several ways in which the current user is helped with information derived from the behavior of other community members.
*(The eyes icon indicates that a result has been promoted to a higher position in the result list than it would have had otherwise.)*

and the tendency of users to be influenced by the fact that particular other users have already expressed a degree of interest in a given link.

## 2.3 Ideas for Existing Group Recommenders

One goal of this chapter is to present a set of issues, concepts, and examples that will stimulate thinking about new group recommender systems. As illustrations, we now give examples of how the design of existing systems might be enhanced with ideas from other systems.

Although collaborative preference specification has not been employed in MUSICFX or MOVIELENS, the analysis given above indicates that it might be worthwhile to extend them in this way. Consider, for example, two friends who visit a gym together and are asked to rate the 91 music genres: If they know that they have similar tastes, they might appreciate a convenient way to fill in the form jointly, so that preferences on which they agree need to be specified only once.

Similarly, if $m_1$ and $m_2$ are buddies within the MOVIELENS system, it is likely that their tastes in movies will overlap to a greater extent than the interests of an arbitrary pair of movie-goers. How might $m_1$ benefit from being able to use $m_2$'s ratings as a starting point for her own ratings? The most obvious case would be the one in which many of $m_1$'s ratings coincide exactly with those of $m_2$; but even simply knowing which movies $m_2$ has rated at all might be helpful: Perhaps the most tedious thing about using MOVIELENS is the job of finding movies that you have seen and so are able to rate—a process that may require scrolling through a list of movies that extends over many web pages. A list containing the set of movies that $m_2$ has rated but $m_1$ has not rated might contain a higher proportion of movies that $m_1$ will be able to rate.

# 3 Aggregating Preferences

## 3.1 Approaches to Preference Aggregation

Even if a group recommender does not elicit members' preferences explicitly, it must have some information concerning the various users' preferences. The recommendation for the group will then in general be based on information about the preferences of all of the group members. Therefore, some type of *aggregation* method is required. The need to choose such a method is the most obvious and intensively studied difference between group recommendation and recommendation for individuals. The topic of preference aggregation is a multifaceted and complex one that has been addressed in various scientific fields (see, e.g., Arrow [11], for a seminal contribution and Masthoff [12] for a summary of this literature from the perspective of group recommendation).

The most general point that we will make about this problem is that a number of complex issues are involved, which take a different form from one system and setting to the next and which therefore make it impossible to formulate a general solution that is satisfactory for all cases. As a way of introducing these issues, we will first summarize how the aggregation problem has been dealt with in our example systems.

Although the various approaches differ in the ways in which they gather and represent users' preferences, almost all approaches make use of one of two schemas: aggregation of ratings for individuals and aggregation of preference models.

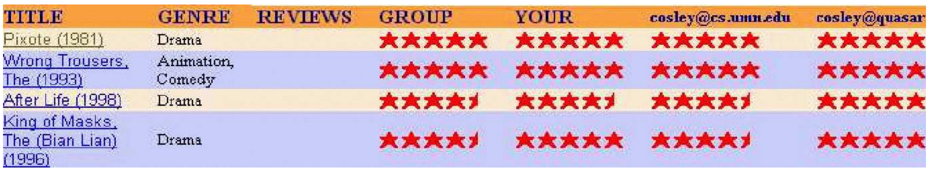
**Aggregation of Ratings for Individuals**

The first approach starts with the assumption that, for each candidate item $c_i$ and each member $m_j$, the system can predict how $m_j$ would evaluate (or *rate*) $c_j$ if s/he were familiar with it:

1. For each candidate $c_i$:
   - For each member $m_j$ predict the rating $r_{ij}$ of $c_i$ by $m_j$.
   - Compute an aggregate rating $R_i$ from the set $\{r_ij\}$.
2. Recommend the set of candidates with the highest predicted ratings $R_i$.

This approach is illustrated by POLYLENS, as can be seen in Figure 4. The three right-hand columns in the screen shot display ratings that have been predicted for individual users via the same collaborative filtering method that MOVIELENS uses for individual users. The column labeled "GROUP" shows the aggregated rating. In POLYLENS, the aggregation method is very simple:

$$R_i = \min_j r_{ij}. \tag{1}$$



| TITLE | GENRE | REVIEWS | GROUP | YOUR | cosley@cs.umn.edu | cosley@quasar |
|---|---|---|---|---|---|---|
| Pixote (1981) | Drama | | ★★★★★ | ★★★★★ | ★★★★★ | ★★★★★ |
| Wrong Trousers, The (1993) | Animation, Comedy | | ★★★★★ | ★★★★★ | ★★★★★ | ★★★★★ |
| After Life (1998) | Drama | | ★★★★½ | ★★★★½ | ★★★★½ | ★★★★★ |
| King of Masks, The (Bian Lian) (1996) | Drama | | ★★★★½ | ★★★★★ | ★★★★½ | ★★★★★ |

Fig. 4. Example of a display of group recommendations in POLYLENS.
*(Image taken from the* GROUPLENS *web site [Reference will be made more precise]. Explanation in text.)*

That is, instead of looking for the movie with the highest average rating, POLYLENS applies the strategy of "least misery": It bases its recommendation on the lowest predicted rating for each candidate, preferring candidates for which the lowest predicted rating for any group member is relatively high. Other plausible aggregation methods will be mentioned in Section 3.2.

### Aggregation of Preference Models

The second general approach to aggregation does not involve any predictions for individual users. Instead, the system uses some method to arrive at an aggregate model for the preferences of the group as a whole:

1. Compute an aggregate preference model $M$ that represents the preferences of the group as a whole.
2. For each candidate $c_i$, use $M$ to predict the rating $R_i$ for the group as a whole.
3. Recommend the set of candidates with the highest predicted ratings $R_i$.

With regard to Step 1: There are even more possible aggregation methods for the aggregation of models than for the aggregation of individual ratings, since models can take many different forms. An example is given by LET'S BROWSE: Each individual user's profile is a set of keyword/weight pairs that reflects the typical content of the pages that this user likes to view. The system computes a model of the group by forming a linear combination of these individual models. From then on, it no longer has to consult the individual models when making recommendations.

INTRIGUE likewise works with a group model, and in fact it never constructs individual models in the first place. Instead, the tourist group leader divides the tour group into several categories of homogeneous users and specifies a preference model for each such subgroup. The group model is then a weighted average of the subgroup models, with the weights reflecting the importance of the subgroups (e.g., the subgroup of disabled persons is considered especially important because of the special requirements of its members).

Although the "groups" for which MUSICFX chooses music change from one moment to the next, MUSICFX does compute a sort of group model for the set of people that are working out in the gym at any given moment, changing the group model whenever someone arrives or leaves. To compute the group preference for any given genre of music, it in effect computes the average of the squares of the ratings of that genre by the people who are currently working out. Whenever a new song is to be played, this group model is used to select the genre of the song. But the system does not simply choose the genre with the highest overall rating: For the sake of fairness and variety, it chooses randomly among the most popular genres.

The TRAVEL DECISION FORUM takes the focus on group models one step further: In fact, the main function of the system is to help the group members arrive, for each aspect of the vacation that the group members are planning, at an aggregate group model that all members have agreed to—that is, at a way of filling out each preference specification form (such as the one shown in Figure 2) in such a way that it reflects the preferences of the group as a whole. If we look at the system in this way, the system can be seen as a system that recommends specific preferences for the group model (e.g., a rating of ++ for the attribute "Sauna" in Figure 2).

I-SPY likewise creates a group (or community) preference model, and in fact it does not create any individual preference models in the first place (partly because of privacy considerations, as will be discussed shortly). I-SPY's basic community preference model consists of a record of queries that have been submitted (by the community of searchers) and the result pages that have been selected for these queries, along with frequency information for these selections. When deciding to what extent to promote a particular search result for a particular community, I-SPY bases its decision on an estimate of how relevant this result page is likely to be for the current query. This estimate is based on the frequency with which this page has been previously selected by community members for the current query and for similar queries.

### Choosing Between the Two Broad Approaches

Constructing a preference model for the group has the clearest advantages when the group members will have an opportunity to examine and/or negotiate about the group's model before or after it is actually applied. In this case, for example, the users of INTRIGUE or the TRAVEL DECISION FORUM, could settle among themselves once and for all the relative priorities of historical interest and entertainment, instead of debating this issue with respect to each individual attraction. This type of process will be discussed further in Sections 4 and 5.

If, on the one hand, the group model will be created and applied in the background, without inspection by the group members, the question of whether a group model is better is a more technical one that involves considerations such as efficiency and the quality of recommendations. For example, O'Connor et al. [2] discuss various ways in which POLYLENS could have been designed to create a model of each group (e.g., a "pseudo-user" who represents the interests of the group as a whole) before any recommendations were generated—and some typical consequences of such group models. For instance, a group model might (accurately or not) recommend a movie for which the predicted rating of each individual member was low—something that cannot happen with recommendation-level aggregation.

Another advantage of a group preference model concerns its potential privacy benefits. Recording and maintaining individual user profiles will typically raise privacy concerns, especially if these profiles are owned by some third-party system on the server side. In contrast, the use of a group preference model may go a considerable way toward alleviating these privacy concerns. I-SPY is a case in point. Our web search behavior can be surprisingly revealing when it comes to understanding the likes and dislikes of an individual—far more revealing and valuable than movie or music preferences, for example. I-SPY's use of a community-based profile, in which the search behavior of individual searchers is merged, means that the search preferences of any individual searcher can no longer be reconstructed.

## 3.2 Alternative Goals and Procedures for Aggregation

Even once a general approach has been chosen, the question arises of what particular computational procedure (or *mechanism*) should be used for the aggregation. This is the single question in this area that has received the most attention. The problem is that there are a number of goals that may be desirable in any given situation and conflicts between them can easily arise. In this section, we give several important examples of such goals—for concreteness, assuming that the overall approach taken is the aggregation of predictions for individuals.[3] Whereas may treatments of these issues (see, e.g., Masthoff [12]; Yu et al. [13]) devote considerable attention to mathematical formulas, quantitative examples, and technical concepts, we will focus on the basic underlying issues and concepts and how they relate to realistic application scenarios. We claim that the first step in designing a solution should be to find a good match between the basic approach taken and the demands of the application scenario; technical considerations come into play when the designer is looking for the local optimum within each basic approach.

**Maximizing Average Satisfaction**

This goal can be achieved by an aggregation function that computes some sort of average of the predicted satisfaction of each member for use as a basis for the selection of candidates (see Equation 2). If the predicted ratings are not thought to represent satisfaction accurately, some transformation of them can be used, such as the square of the rating. [4]

$$R_i = \text{average}(\{r_{ij}\}) = 1/n \cdot \sum_{j=1}^{n} r_{ij}. \tag{2}$$

**Minimizing Misery**

Even if the average satisfaction is high, a solution that leaves one or more members very dissatisfied is likely to be considered undesirable. Even the most ego-centered group member may not want to have to interact with another member who is thoroughly dissatisfied; and such a member may refuse to go along with the solution in any case. In POLYLENS, the minimization of misery is the only criterion applied (see Equation 1 above). It is also possible (e.g., as in an early version of MUSICFX) to take this factor

[3] As was noted above, where preference models are being aggregated, the variety of possible aggregation methods is much greater, and accordingly the possible ways of taking these goals into account are more diverse.

[4] A transformation of this sort is performed on the level of models in MUSICFX (cf. Section 3.1).

into account as a constraint that must be fulfilled by a solution: The lowest predicted rating must not fall below a given threshold.

**Ensuring Some Degree of Fairness**

In a similar vein, a solution that satisfies everyone just about equally well is in general preferred to one that satisfies some at the expense of others—all other things being equal. Even more than in the case of minimizing misery, the goal of ensuring fairness is in general combined with some other goal. After all, no-one wants a perfectly fair solution that makes everyone equally miserable. For example, the aggregation of predicted individual ratings might include a penalty term that reflects the amount of variation among the predicted ratings, as in Equation 3:

$$R_i = \text{average}(\{r_{ij}\}) - w \cdot \text{standard-deviation}(\{r_{ij}\}), \tag{3}$$

where $w$ is a weight that reflects the relative importance of fairness.

**Discouraging Manipulation of the Recommendation Mechanism**

The issue of manipulation is illustrated by experience with an early version of MUSICFX: As part of the aggregation method sketched in Section 3.1, the system enforced a constraint of the type mentioned above in connection with the "least misery" criterion: Any music genre that was "hated" by any member currently in the gym was removed from the list of possible genres to play. Some users were observed to force an immediate change of genre by adapting their specifications to indicate that they "hated" the genre currently being played—even if they really didn't mind it but simply liked it less well than some other genres.

The potential for manipulation is even more obvious in the TRAVEL DECISION FORUM, in which one group member can often see the preferences specified by the other members. For example, suppose that in Figure 2 Claudia's true preference regarding the presence of a sauna was $\sim$ ("Don't care"): Instead of selecting the middle box in the scale, she might be inclined to select the left-most box (indicating strong disapproval of the availability of a sauna), so as to compensate for the positive preferences specified by the other group members Ritchie and Tina, expecting that the aggregated group preference for a sauna will end up being closer to her own.

When this type of insincere specification of preferences occurs, the aggregation algorithm used will be operating on false premises, since the algorithms assume that a group member's expressed preferences reflect his or her true preferences. Approaches to this problem will be discussed below in Section 3.3.

**Ensuring Some Degree of Comprehensibility**

As will be discussed in Section 4, group members sometimes like to be able to understand the rationale behind a recommendation. In particular, they may want to check to what extent criteria such as the ones just listed are being fulfilled. The median mechanism mentioned below in Section 3.3 is an example of one that is easy even for unsophisticated users to understand, apply, and remember. By contrast, automatically designed mechanisms (again, see Section 3.3) tend to be much less comprehensible, unless special constraints are specified that are intended to enhance comprehensibility.

#### Treating Different Group Members Differently (Where Appropriate)

In some situations, it is generally agreed that the evaluations of some group members need to be treated differently than those of others. If two hosts are planning a visit to a restaurant with a visitor from out of town, they are likely to give high priority to the visitor's preferences, requiring only that the solution is not entirely unsatisfactory for themselves. In INTRIGUE, the tourist guide is able to assign higher weights to subgroups such as disabled persons or children, on the assumption that these group members are less able to put up with solutions that are even partly unsatisfactory for them. Some aggregation procedures make it easy to treat different group members differently. For example, a weighted averaging procedure can make use of weights that correspond to the relative importance of each member.

### 3.3 Strategies for Combating Manipulation

One way of making manipulation difficult is to make it impossible for users to see each others' preferences before specifying their own: If you don't know what the other members prefer, it is hard to distort the resulting recommendation in your own direction by specifying an insincere preference. But users may be able to guess other members' preferences (at least roughly); and in any case, as we have seen, there are advantages to allowing members to see each others' preferences at an early stage.

Manipulation is most likely to be possible if the input that the system uses for making its predictions consists of explicit preference specifications; with implicit inference of preferences, users are much less likely to be able to see how they could influence a recommendation by acting in some particular way. But exceptions can occur; for example, in I-SPY, user can quickly notice that, when they choose a given link for a given query, that link gets promoted in the search result list for that query. It is then an obvious next move to click on links that one would like to promote (e.g., pages written by the user), regardless of their actual relevance to the query. One can view this type of manipulation as an alternative form of search engine spam, because subsequent users will see potentially irrelevant results being unjustifiably promoted to positions of prominence. As a potential solution to this problem Briggs and Smyth [14] propose the use of an explicit model of trust that provides a filtering mechanism with a view to eliminating the contributions of these manipulative selections: The selections of individual users are evaluated for their reliability. In the simplest sense, a result selection is considered to be reliable if the same link is subsequently reselected by a certain minimum number of searchers for similar queries in the future. This information is used for (among other things) the evaluation of the trustworthiness of individual users, so that recommendations that stem from the activities of users with low trust values can be eliminated or demoted. Preliminary evaluation results suggest that the technique is capable of improving recommendation accuracy.

A different approach to discouraging manipulation is to have the system use an aggregation method that is inherently *nonmanipulable*: It is never in the interest of a given user to specify any preference other than the one that he or she really has. To return to the example with the TRAVEL DECISION FORUM given above: A simple nonmanipulable aggregation mechanism uses as a preference for the group as a whole the *median* of the individual preferences (i.e., the one that falls exactly in the the middle of an ordered list of all preferences). In our example, Claudia will not be able to drag down the group preference for a sauna below $+$ by specifying a low preference herself, since the median preference will be $+$ for any preference that she specifies between $--$ and $+$. (It will be left as an exercise for the reader to verify that, with the use of the median mechanism in this setting, no group member could ever benefit by specifying a preference insincerely.)

In general, many nonmanipulable mechanisms may exist for any given preference aggregation problem. The recently initiated research area of *automated mechanism design* (see, e.g., Conitzer and Sandholm [15], Conitzer and Sandholm [16], Jameson et al. [17]) develops ways of automatically generating such mechanisms that respect other constraints as well (e.g, maximizing average satisfaction and/or ensuring a certain degree of fairness.) The TRAVEL DECISION FORUM allows an administrator or a user to specify the properties that an automatically designed mechanism should have, and it then generates the mechanism on the fly.

## 3.4 When Can an Appropriate Procedure Be Chosen?

The tendency of the goals listed in Section 3.2 to conflict can be dealt with on three different levels:

1. *By designers, before the system is deployed:* The persons who are designing a group recommender—or arranging the deployment of an existing recommender in a given context—can consider how important each of these goals is for their particular target group and application setting so as to work out some appropriate solution. For example, the designers of POLYLENS thought that the "least misery" aggregation function would be appropriate because they expected most groups of people who go to see a movie together to be small (i.e., 2 or 3 members); for settings involving larger groups, this same function would probably lead to too many cases in which a solution that would be liked by many members would in effect be vetoed by the one person who liked it least. Similarly, in some settings manipulation may not be an issue (e.g., because preferences are not explicitly specified but rather inferred on the basis of behaviors, such as purchasing decisions, which people are unlikely to distort intentionally in order to manipulate a recommender); in these cases, the designer need not worry about the goal of avoiding manipulation.
2. *By users, when an aggregation function is selected:* A system can allow the users to decide what aggregation mechanism is to be used, either before any recommendations are made or during an iterative process of requesting recommendations and adjusting the aggregation function. For example, with INTRIGUE the tour guide can specify a different set of subgroup weights for each tour group. As was mentioned above, a variety of aggregation mechanisms can be chosen in the TRAVEL DECISION FORUM.

3. *By users, when specific recommendations are being considered:* If the system presents a number of recommendations and allows users to choose which one(s) they want to adopt, it may not be necessary for the system to take the various goals into account in an optimal way. Instead, the users themselves can take the goals into consideration, weighting them appropriately, when they make their final decisions. The system should, however, take the goals into account well enough so that the set of candidates offered is likely to include one or more highly suitable options.

## 3.5 Situations Where Recommendations Are Made Concerning Multiple Decisions

So far, we have been implicitly focusing on the situation in which a recommender will make recommendations to a group concerning just one decision. But often the group members will expect a system to make recommendations concerning a larger set of decisions, either at the same time or in succession: INTRIGUE's tour guide will choose several sights to visit; LET's BROWSE will recommend a number of web pages in the course of a given session; and a TV program recommender will recommend several programs for a group to watch in succession.

In this type of situation, an even broader range of aggregation procedures is available. For example, Yu et al. [13] discuss (among others) a simple aggregation procedure that avoids the construction of a group model but does not require the aggregation of predicted ratings for individuals (cf. Section 3.1): The system generates a recommended sequence of programs for each group member individually and then constructs a merged sequence for the group by choosing programs from these individual sequences.

Moreover, the goals discussed above may need to be considered with regard to the whole set of decisions rather than with regard to single decisions. Consider, for example, the goal of avoiding extreme cases of dissatisfaction: It may seem all right to make the children in a tour group suffer through one or two attractions that are of interest only to adults—as long as the overall satisfaction of the children with the sightseeing tour is reasonably high. Trying to avoid "misery" for each participant at each sight visited might rule out too many options that were attractive for the group as a whole.

Similarly, the criterion of fairness can be applied either locally (to each individual decision) or globally (to the entire set of decisions). The former approach can lead to a sequence of decisions none of which satisfies any member very much, while the latter approach allows each member to "get their way" some of the time

Consideration of an entire sequence of decisions is most straightforward when the decisions are made at the same time, as a package (as is the case in Figure 2, where the TRAVEL DECISION FORUM recommends six group preferences at once). It is harder when the decisions are handled one by one: At any given time before the last decision is considered, neither the system nor the group members may know what decisions remain to be made in the future.

# 4 Explaining Recommendations

## 4.1 Motivation

Given the many ways in which recommendations for a group can be derived—and the often conflicting goals that can be pursued—it is natural that group members should want to understand to some extent how a recommendation was arrived at—and in particular, how attractive a recommended item is likely to be to each individual group member.

Recommender systems for individuals often accompany each recommendation with some sort of analysis of its predicted acceptability; the analysis may range from a simple index of the system's confidence to a complex visualization of the pros and cons of the recommended solution (see, e.g., Herlocker et al. [18], and the chapter in this volume by Herlocker et al.). With group recommenders, it is in principle possible to present such an analysis for each individual member, for the group as a whole, and perhaps for subsets of members. A member $m_1$ may be interested in the analysis for $m_2$ because $m_1$ considers it important that $m_2$ be satisfied, because $m_1$ wants to make sure that she is getting "as good a deal" as $m_2$, or simply in order to understand how the recommendation was derived.

## 4.2 Treatment in Existing Group Recommenders

As can be seen in Figure 5, LET'S BROWSE explains each of its web page recommendations by listing the keywords in the page that it assumes to be of interest to all group members. By showing where these keywords are located in each member's profile, the system also allows each user to guess how interesting each member will find the page. In the example in the figure, it looks as if the (hypothetical) group member George Lucas will be less enthusiastic about the page than the member Bill Gates, given that Lucas is only marginally interested in technology. As some of the systems to be discussed below suggest, it might be a worthwhile further step for LET'S BROWSE to present an explict estimate of the likely interestingness of the page for each group member and perhaps for the group as a whole. In this way, for example, Lucas might more readily accept the system's recommendation of the page involved in Figure 5, seeing quickly that it is more interesting to other members than it is to him; or, depending on his overall attitude, he might object to the recommendation for just that reason. On the other hand, since (as was mentioned in Section 3.1) LET's BROWSE uses a group-level model to compute its recommendations, the computation of predictions for individual group members for presentational purposes would involve additional overhead, and it would not reflect the way in which the system actually arrived at its recommendation.

POLYLENS gives an idea of how a display that shows predictions for individuals might look. Since POLYLENS uses collaborative filtering, it cannot explain a movie recommendation in terms of the movie's content; but it does show the predicted rating for each group member and for the group as a whole (see Figure 4 above). In addition to explaining each recommendation in terms of the underlying predictions for individuals, this visualization makes it possible for the attentive user to notice how group recommendations are generated—via the "least misery" principle (Equation 1)—by comparing

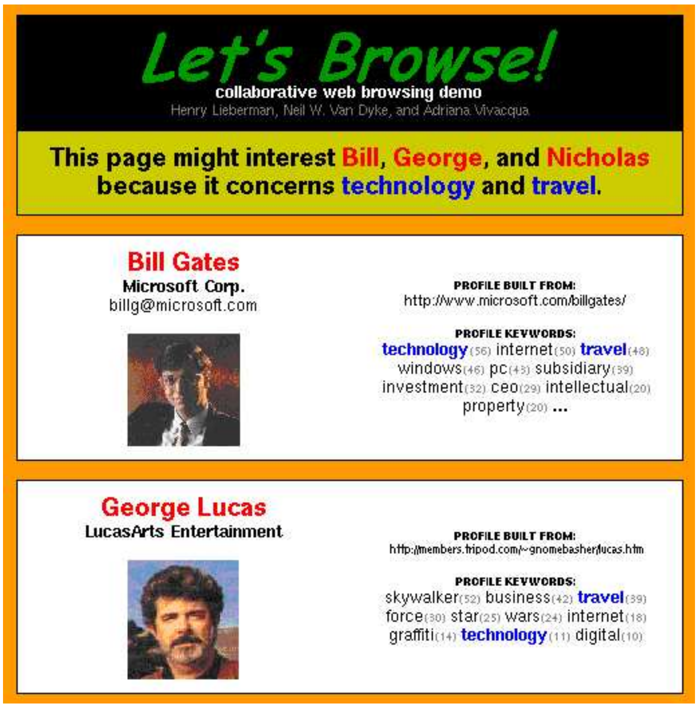



Fig. 5. Screen shot of LET'S BROWSE that shows how the system explains a recommendation for three (hypothetical) users of a particular web page.
*(Adapted from an image taken from Henry Lieberman's web homepage—reference to be made more precise.)*

predictions for the individual members with the predictions for the group. Incidentally, more than 90% of the users surveyed stated that they had no privacy concerns about having their predicted ratings shown to other group members—a result that encourages the development of additional methods that expose individual-level predictions to all group members.

INTRIGUE offers a type of explanation (Figure 6) that is partly similar to that of POLYLENS: It shows the predicted attractiveness of each recommended attraction for the tourist group as a whole; but instead of simply presenting a predicted attractiveness for each subgroup, it explains verbally the aspects of the attraction that are likely to appeal to that subgroup (not mentioning the less appealing aspects). This type of explanation seems helpful for the tour guide who would like all group members to accept the recommendation, but it does not convey a clear idea of how attractive the recommended tour is to each subgroup.[5]

[5] INTRIGUE also offers a different type of explanation that shows only predictions for individual subgroups; see Figure 8 of Ardissono et al. [3].


Fig. 6. Example of INTRIGUE's main explanation method.
*(Adapted from Figure 3 of Ardissono et al. [3].)*

I-SPY incorporates a number of strategies for making clear the reasons for the promotion of a given link in a search result list. Since the users of I-SPY are not viewed as working together when they look for relevant links, the focus here is not on showing the desirability of an option for particular group members with a view to resolving conflicts. Nonetheless, it has proven worthwhile to provide information about how other community members have dealt with the page in question in the past because such information helps the current user to judge its value for him- or herself. Types of information offered include (a) *related queries* for which the page in question has been selected as a promising result (see Figure 3); (b) quantitative and temporal information such as "10% of searchers have also selected this result for similar queries as recently as 15 minutes ago" (Coyle and Smyth [19]); and (c) the names of the users who are responsible for the promotion of the page (a by-product of the anti-manipulation measures discussed in Section 3.3; see Briggs and Smyth [14]).

The TRAVEL DECISION FORUM introduces two novel, complementary methods that aim to provide a more detailed picture of the consequences of a given proposal for each group member:

1. The first method automatically follows from the use of the preference specification form for the presentation of proposals (see Figure 2). Since both the specified preferences and the recommended joint preferences are shown on the same set of scales, the user can quickly see which group members should be most / least satisfied with a given proposal (i.e., the ones whose preferences are closest to / farthest from the highlighted cells). Also, with a bit of practice the user can see more complex patterns (e.g., Claudia might notice that "Tina and Ritchie have generally similar preferences, and they usually get their way, while my preferences have little influence"). Any verbal arguments


Fig. 7. Snapshots of some reactions of the representative of Tina to the proposal shown in Figure 2.

associated with the other members' stored preferences add further detail to the picture of how they would evaluate a given proposal.

2. The second method takes into account the fact that any graphical explanation of a recommendation is likely to be less interesting, vivid, and memorable than the type of feedback that group members get while they are interacting face to face: A member who is disappointed with a proposal may complain about specific aspects of it in an emotional manner, formulating (or repeating) arguments. In settings where all group members are physically present in front of the group recommender system, this type of face-to-face discussion is likely to occur spontaneously. For settings in which no such direct communication is possible, the TRAVEL DECISION FORUM tries to recapture some of the flavor of face-to-face interaction through animated characters: It is assumed that at any given moment only one group member will be interacting with the system; each of the other members is represented by an animated character who bears that member's name. Whenever the system (represented by an animated character called the *mediator*) has recommended a particular joint preference model for a given value dimension (such as "health facilities"), he asks the representatives of the absent group members to comment on it in turn. Parts of a typical performance of a representative are shown in Figure 7.

This type of simulated reaction can heighten the group members' awareness of the other members' points of view—including their motivational orientations—and overcome the natural tendency to focus on one's own evaluations. Also, like the explanations of INTRIGUE, these presentations are selective, focusing on the most important considerations for each group member. The user can switch attention back and forth between the animated agents and the graphical explanation, because the two types of explanation make use of largely complementary communication channels.

# 5 Helping Group Members to Arrive at a Final Decision

No matter how appropriate and compelling a system's recommendations and explanations are, there is usually no guarantee that any of the recommendations will be adopted. With individual recommenders, although the decision process may be complex, it typically takes place within the mind of a single person. With a group recommender, extensive debate and negotiation may be required, which may be especially problematic if the members are not able to communicate easily.

## 5.1 Treatment in Existing Systems

Group recommender systems have tended to avoid the issue of final decision making in various ways, which mostly make good sense given the application scenarios:

1. The system simply translates the most highly rated solution into action without requesting the consent of any users.
   This method is applied by MUSICFX, which switches music channels autonomously on the basis of the preferences of the group members who are present. It would in fact be impractical to allow the persons working out in a gym to debate about each change of music genre.
2. It is assumed that one group member is responsible for making the final decision.
   LET'S BROWSE is based largely on the assumption that one group member controls the pointing device and will therefore make most of the decisions about what pages to visit. If any debate arises, it can be handled through face-to-face discussion. With INTRIGUE, it appears to be assumed that the tourist guide will decide what tour should be taken. Since the individual members of the group do not access the web-based recommender, any debate has to take the form of face-to-face discussion with the tour guide.
3. It is assumed that group members will arrive at the final decision through conventional discussion (e.g., face-to-face or by phone).
   This assumption may be reasonable for a system like POLYLENS if all group members can easily communicate when the system makes and explains its recommendations. Even in this relatively favorable case, effective and accurate representation of the consequences of particular solutions for individual users can help to streamline the decision making process.

The only system we know of that explicitly supports communication among group members for the purpose of final decision making is the TRAVEL DECISION FORUM—understandably, since this system is designed specifically for groups of users who usually cannot communicate with each other in real time. The animated representatives of absent group members (Section 4) do not serve only as a means of visualizing the implications of recommended solutions for the absent members. In addition, each member can grant her representative a certain amount of authority to accept proposals during interactions with another group member. For example, in Figure 7, Tina's representative states that she cannot accept the current proposal. If instead the representative had accepted it and the same were true of Ritchie's representative and the current user Claudia, the proposal would have been treated as finally accepted, even though Tina and Ritchie had not really seen it.

### 5.2 Possible Extensions to Existing Systems

Since in most applied scenarios the overhead of animated agents would be too great, we should consider how functions such as those of the TRAVEL DECISION FORUM can be realized in a lighter-weight manner. For example, a system like POLYLENS might allow each member to specify that they are willing to go to any movie whose predicted rating for them is above some threshold (e.g., 4 stars out of 5). In that case, the system could present not only recommendations but also a subset of the recommended movies that can be decided on without further consultation; the designated group leader could then go on and buy tickets for any of these movies. If it is assumed that each group member will view the recommendations before a decision is made, a procedure could be introduced that allowed the group members to vote for movies among the recommended ones, also indicating which particular ones they are willing to accept. In this case as well, it could be agreed that a designated group member could make the final decision.

As the reader may have noticed, this type of voting mechanism can in itself be viewed as a simple recommender system that makes use of explicit preference specifications and helps the group members to choose among the recommended options. And in fact we could apply all of the concepts introduced so far to this "recommender", considering, for example, whether the group members should be allowed to see each other's votes (cf. Section 2), how the votes should be counted and weighted (Section 3), how the results of the voting should be presented (Section 4), and even how the really final decision ought to be made (the present section). Fortunately, we are not faced with an infinite regress, since the recommendation problem that we are dealing with now—choosing from among a small number of recommended items—is considerably simpler than the original problem, and simple solutions may be quite adequate. For example, suppose that for the original recommendations an aggregation function was used which gave especially high weight to particular group members (eg., the visitors from out of town). It may not be necessary for the final voting mechanism to be biased in their favor as well, since the set of recommendations itself will already contain mainly films that the guests will like; and the hosts are likely in any case to vote in a way that takes into account the greater importance of the guests.

More generally, decisions are often made in several stages, and in each stage a different type of decision making procedure may be applied. Issues that require a great deal of attention in one stage may be simpler to deal with in another stage.

## 6 General Conclusions

### 6.1 Conclusions Concerning Group Recommender Systems

So far, only a small number of recommender systems have been designed for groups. The recommendation techniques that they have employed represent only a small subset of the possible recommendation techniques, most of which are discussed in other chapters in the present volume (those of Pazzani, by Smyth, by Burke, and by Goy et al.). And of course the application domains that we have seen in this chapter represent only a limited sample. As group recommendation becomes more widespread, the general issues discussed here will arise again and again in different forms and contexts, and the solutions generated will constitute a growing "case base" from which the designers of yet newer systems can learn. But even within the limited sample of systems that we have looked at here, we have seen a number of cases in which ideas from one system can be transferred to another system, even if the two seem quite different superficially, as long as we keep in mind the common general issues. In particular, a design solution that seems fairly obvious for one particular system may in fact also be applicable—with adjustments—to a very different type of system for which it is not at all obvious.

At the same time, we have seen that the best way of handling each issue can depend strongly on the nature of the system in question and its application scenario. That is, the adoption of design solutions from other systems calls for a willingness to take into account creatively both large and subtle differences.

### 6.2 Implications for Other Types of Adaptive Web-Based System

Even more generally, just about any type of system that adapts to its users can be seen in some sense as a recommender system, and it may be expandable for adaptation to groups of users. For example, a system for personalized information access (cf. the chapter by Gauch, Speretta, and Micarelli) can be seen as "recommending" particular documents to a user; and it is reasonable to adapt to groups of cooperating, information-seeking users. Either explicitly or implicitly, we will then have to deal with issues such as the specification and aggregation of preferences of the various group members.

Similarly, a system that offers adaptive navigation support (cf. the chapter by Brusilovsky) can be seen as recommending moves within a hyperspace; and it is natural to consider groups of navigating users (as is in fact done in LET'S BROWSE).

Therefore, we hope that the analytical framework, the examples, and the general concepts that we have presented in this chapter will be found helpful by designers of a broad range of adaptive web-based systems.

## 7 Acknowledgments

For their support for the preparation of this chapter and their own relevant research, the authors are grateful to the following sources: for Anthony Jameson, the German Ministry of Education and Research (BMBF) (projects MIAU and SPECTER); for Barry Smyth, Science Foundation Ireland under grant 03/IN.3/ I361 and the Informatics Research Initiative of Enterprise Ireland.

## References

1. Lieberman, H., Van Dyke, N.W., Vivacqua, A.S.: Let's Browse: A collaborative Web browsing agent. In Maybury, M.T., ed.: IUI99: International Conference on Intelligent User Interfaces. ACM, New York (1999) 65–68

2. O'Connor, M., Cosley, D., Konstan, J., Riedl, J.: PolyLens:A recommender system for groups of users. In: Proceedings of the European Conference on Computer-Supported Co-operative Work. (2001)
3. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: INTRIGUE: Personalized recommendation of tourist attractions for desktop and handset devices. Applied Artificial Intelligence **17** (2003) 687–714
4. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: Proceedings of the 1998 Conference on Computer-Supported Cooperative Work. (1998) 363–372
5. Jameson, A.: More than the sum of its members: Challenges for group recommender systems. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, Gallipoli, Italy (2004) 48–54
6. Jameson, A., Baldes, S., Kleinbauer, T.: Two methods for enhancing mutual awareness in a group recommender system. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, Gallipoli, Italy (2004)
7. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction: The Journal of Personalization Research **14** (2005) 383–423
8. McCarthy, J.F., Anagnost, T.D.: MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: CSCW 2000 Video Program. ACM, New York (2000)
9. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. International Journal of Group Decision and Negotiation **10** (2001) 199–215
10. Smyth, B., Balfe, E., Boydell, O., Bradley, K., Briggs, P., Coyle, M., Freyne, J.: A live-user evaluation of collaborative web search. In Kaelbling, L.P., Saffiotti, A., eds.: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco (2005) 1419–1424
11. Arrow, K.J.: Social Choice and Individual Values. 2nd edn. Wiley, New York (1963)
12. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modeling and User-Adapted Interaction (2004)
13. Yu, Z., Zhou, X., Hao, Y., Gu, J.: TV program recommendation for multiple viewers based on user profile merging. Manuscript submitted for publication, Northwestern Polytechnical University, People's Republic of China (2006)
14. Briggs, J., Smyth, B.: [title to be supplied in next version]. In: Proceedings of AICS 2005. (2005) Full reference to be supplied.
15. Conitzer, V., Sandholm, T.: Complexity of mechanism design. In Darwiche, A., Friedman, N., eds.: Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference. Morgan Kaufmann, San Francisco (2002) 103–110
16. Conitzer, V., Sandholm, T.: Applications of automated mechanism design. Unpublished manuscript, available from http://www-2.cs.cmu.edu/∼conitzer/ (2003)
17. Jameson, A., Hackl, C., Kleinbauer, T.: Evaluation of automatically designed mechanisms. In: Proceedings of the First Bayesian Modeling Applications Workshop at the Nineteenth Conference on Uncertainty in Artificial Intelligence, Acapulco, Mexico (2003)
18. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 Conference on Computer-Supported Cooperative Work, Philadelphia, PA (2000) 241–250
19. Coyle, M., Smyth, B.: Explaining search results. In Kaelbling, L.P., Saffiotti, A., eds.: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Francisco (2005) 1553–1555