The Visual Decision Maker – A Movie Recommender for Co-located Users

Tim Regan,
Steven Drucker,
Markus Lofstrom, and
Asta Glatzer

October 4, 2002

Technical Report
MSR-TR-2002-77

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

The Visual Decision Maker – A Movie Recommender for Co-located Users

Tim Regan, Steven Drucker, Markus Lofstrom, and Asta Glatzer
Microsoft Research,
1 Microsoft Way
Redmond, WA 98052 USA
+1 425 706 0585
{timregan, sdrucker, mlofstro, astag}@microsoft

ABSTRACT
We present the Visual Decision Maker (VDM), an application that gives movie recommendations to groups of people sitting together. The VDM provides a TV like user experience: a stream of movie stills flows towards the center of the screen, and users press buttons on remote controls to vote on the currently selected movie. A collaborative filtering engine provides recommendations for each user and for the group as a whole based on the votes. Three principles guided our design of the VDM: shared focus, dynamic pacing, and encouraging conversations. In this paper we present the results of a four month public installation and a lab study showing how these design choices affected people's usage and people's experience of the VDM. Our results show that shared focus is important for users to feel that the group's tastes are represented in the recommendations.

Keywords
Movies, collaborative filtering, user modeling, shoulder to shoulder, single display groupware, co-located collaboration, multi-person interfaces, multiple input devices, flow, awareness

INTRODUCTION

The Visual Decision Maker (VDM) is a visually compelling movie recommendation systems designed for individuals or co-located groups of users. It integrates multiple modules:

- Front end – a stream of images with input facilities and associated feedback
- Back end – a user modeling system from which recommendations are made
- Database – the store of media and associated meta-data
- Media – images (stills from movies) used in the front-end stream and referenced from the database.

The visual decision maker acts by presenting users a stream of movie stills, on which users express their like or dislike. Using these responses, the collaborative filtering engine builds up a model of preferences for the users. Given this model, the computer generates a list of recommendations. By doing this simultaneously with multiple users, in a dynamic fashion, the system is useful, fun, and instigates conversation among the participants. A version of the VDM interface is shown in Figure 1.

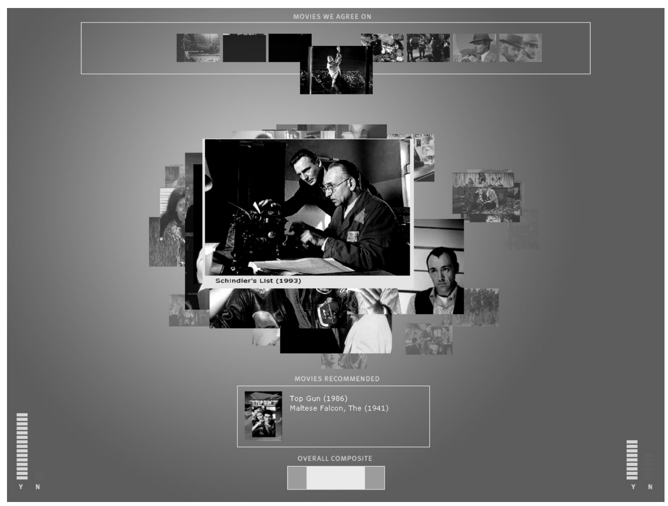

Figure 1: The Visual Decision Maker Screen with the flow of images in the center, agreed upon images at the top of the screen, recommended movies in the central box, and feedback indicators showing amount of agreement and a history of each user's votes

We report the results of two experiments in this paper. In the first, the VDM was installed in one of the cafes on a large corporate campus. We analyzed users' reactions to the system and did quantitative analysis of their sessions' characteristics. In the second, we tested the VDM with pairs of users in a lab setting experimentally explore our design choices. Based on these experimental results this paper includes design recommendations for others building group applications for co-located users. Our finding that the perceived quality of the recommendations can be influenced through UI design, rather than just through accuracy of the recommendation system, is important for any application that includes group suggestions, especially those gleaned from collaborative filtering.

RELATED WORK

Most recommendation systems are designed for solitary users. Notable exceptions include Flytrap [2] and MusicFX

[5] from the intelligent room literature, and PolyLens [10]. The FlyTrap group music environment is a system of software agents that vote to set the music playing in a room. Agents learn the musical tastes of users by observing the music they listen to at their personal machine. Users wear RFID tags so that FlyTrap can identify a room's occupants. MusicFX is a system installed in a corporate gym. Users record their genre preferences on joining the gym, and can update their record at any time. Gym users sign into MusicFX by swiping their corporate security badge across a proximity reader. The user is assumed to have left after a time period predicted by observation of workout times. MusicFX changed people's usage of the gym (sparsely attended times had interesting and novel soundtracks) and dramatically reduced complaints to gym staff about the music played (and it had been their number one source of complaints) – the majority were ignored no more and variety was increased. PolyLens is a modification of the MovieLens web based film recommendation system. It allows users to form groups, invite other users into these groups, and then displays movie recommendations to the group as a whole. The group members access the recommendations through the web at their own PC, so it is not shoulder to shoulder, but many of the issues we faced were also faced in PolyLens.

The VDM differs from Flytrap, MusicFX, and PolyLens in a number of ways. The VDM group experience includes the gleaning of user preferences: the users tell the system which movies they liked or disliked whilst sitting side by side at the same screen. To make the VDM effortless to use we have no sign in and hence no history of user's previous choices. We will return to PolyLens and MusicFX later, when discussing our design choices and collaborative filtering algorithm.

It is commonplace for several gamers to play together at a single console and a single screen. Games deal with this multiplicity of user views by either

1. Giving each user their own character (avatar) to center their attention on in the scene (e.g. Fusion Frenzy [7]);
2. Splitting the screen so that each player watches the action in their own frame (e.g. RalliSport Challenge [9]); or
3. Providing time slots, i.e. users take turns playing (e.g. Amped [8]).

TV content is also commonly consumed by several people in the same room. It is this TV style of viewing that we wanted to explore with the VDM. Users sit back from the VDM screen and interact using remote controls. Two clear precursors to our work are the games of Two Way TV [13] and Jellyvision's You Don't Know Jack series [1]. During 1994 and 1995 Two Way TV offered a game in the UK as an interactive overlay to existing TV quiz shows. Subscribers saw multi-choice answers to the quiz show questions overlaid on their screens and could use up to four remote controls to pick different answers. Families would compete amongst themselves and compare their scores to those of the TV contestants. Two Way TV's current games for interactive TV audiences still have multiplayer facilities, though they now rely on service providers to provide set-top boxes allowing multiple simultaneous inputs. Gottlieb has teased out the principles that went into the design of the You Don't Know Jack games in [1]. The 'Jack Principles' Gottleib puts forward are: maintaining pacing, creating the illusion of awareness, and maintaining the illusion of awareness. One key distinction between the VDM and interactive conversation interfaces supporting the 'Jack Principles' is that the VDM does not try to be one of the parties in the conversation, but instead tries to foster conversation amongst users. Hence some of the 'Jack Principles', especially those around the illusion of awareness, are not applicable to the VDM in full.

Groupware researchers have proposed a number of UI innovations to enhance shoulder to shoulder computing. These are mainly confined to meeting rooms, children's applications, and mixed reality systems, though there are general guidelines too. Overviews of shoulder-to-shoulder computing are contained in Stewart et al. [12] and Inkpen et al. [4].

Using image streams to facilitate conversation and make recommendations to several simultaneous participants appears to be a new idea.

VDM DESIGN REASONING

There are many principles we adhered to in designing the VDM. Some are from Gottlieb's Jack Principles [9] and some are our own. In this section we go through the design decisions made with particular reference to the principles of shared focus, dynamic pacing, and encouraging conversations. The initial VDM was designed for two users and that is the version we will use as the basis for the description in this section. It was used by colleagues and demoed at internal company exhibitions. Please refer to Figure 1 and the supporting video for graphical clarification of the interface items discussed here.

Interface Description

In this section we go through the elements of the VDM interface and discuss the design choices made.

Stream of Stills

The most obvious feature of the VDM's interface is the stream of stills moving from the distance at the periphery of the screen towards the center and the foreground. We choose movie stills over a more text centric interface for several reasons. Firstly movie stills are more evocative than a title and secondly people read at very different rates which is hard to accommodate in a shared interface. We also tried DVD covers as they seemed to balance text and evocative images but they gave the VDM an overtly commercial feel and the radically differing design styles of different DVD covers made the VDM jarring.

The movie stills used in the VDM were copied from the CD-ROM Cinemania 97.

Selected Still

When the top still (in the z-order) reaches the front of the screen it is selected automatically. The selected still is differentiated from the others by a blue border and the addition of the movie title below the film. The inclusion of the title was important as an additional aid to movie recognition, especially for obscure movies. We chose to select one movie to be in focus at a time, rather than letting the user select one or several movie stills to act on, so that we maintained the shared focus of users and to provide a TV-like interaction where the media comes to the consumer with little interaction and effort required.

User Input

Once a still is selected users may express an opinion on it, a process we will term voting. Users can vote that they liked the selected movie, that they did not like the selected movie, or they may opt to skip. Skipping may mean that the user did not see the selected movie, did not know the selected movie, had a neutral opinion of it, or that the user just wanted to get on to the next still.

We tried a number of input devices with the VDM: keyboard entry during development, multiple mice (using [6]), multiple game controllers, and infra-red remote controls. Mice were too restrictive in the number of inputs available (i.e. buttons), while implementing on screen buttons would entail too much navigational load on the user. Hence we choose remote controls. We removed the unnecessary remote control buttons and painted the remaining ones to provide a simple mapping between the on-screen help and the remote control.

Immediate user feedback

When a user votes, their vote is immediately represented on screen by a colored bar appearing under the selected image – green to represent a yes vote, red to represent a no vote, and sky blue to represent a skip vote. This enables users to see how each other are voting. In other group recommendation systems this is seen as a problem, because it shows blatant disregard of users' privacy and inevitably influences users who have not yet voted. Indeed we witnessed this behavior in early pilot testing of the VDM where one user decided to vote in opposition to another just to goad him. This is an example where our goal of fostering conversation between users took our design in a novel direction.

Pacing

The timing of the flow of stills in the VDM responded to users actions. If the users all voted on each selected still quickly then the flow sped up. There were several rules governing the speed of flow. If no-one was using the VDM it went into an attractor mode where the selected image remained on the screen for approximately one second. This provided an enticing visual flow to attract users. When in use, the VDM will wait until each user has voted before allowing the selected still to be removed. The wait is set to 60 seconds to give a user ample time to discuss a contentious movie before voting. Finally once all the users have voted the VDM pauses 1 second before removing the currently selected still and selecting the next one to give users time to see each other's vote.

User history

There are several UI artifacts that give users a history of their use of the VDM since this was an important topic for user discussion. Firstly stills that users both liked animate to the top of the screen to a list. They eventually fade from the list as new stills are added but this gives users sufficient additional time to comment on the movies they both liked. We also provided a visual indication of the total amount of yes and no votes each user had given at the side of the screen. Centrally two overlapping lozenges showed the extent the users agreed and disagreed by varying the degree of overlap. The extent users agree and disagree proved a frequent topic of conversation between users and was encouraged by the provision of these features.

Log On

Because we wanted as few barriers to use as possible we provided no log-in to the VDM. The benefit of this was that users could walk up and start interacting with the system immediately. One disadvantage was that we could not build up an increasingly detailed and accurate model of a user's preferences across multiple sessions.

Recommendations

Partly because of our decision to avoid explicit user log-on and partly to support a exhibition demo usage, the VDM initially gave recommendations as it went along. An area towards the center and bottom of the screen contained a list of the five most recent recommendations with a thumbnail of the DVD cover of the most recent film. The decision to use DVD covers here was pragmatic – we had access to a large array of recent DVD cover pictures so we could recommend more recent movies. Because the DVD cover was a small part of the overall UI its design style did not dominate or clash with the rest of the VDM in the way that they had when used in place of the movie stills.

The recommendations themselves were provided by a collaborative filtering engine using data from the University of Minnesota's MovieLens project [11]. The MovieLens project provided us with a subset of their database restricted to votes from their users who had voted on a movie from our collection of stills. The collaborative filtering used the algorithm described by Breese, Heckerman, and Kadie in [1]. In [10] O'Conner et al discuss the different ways in which collaborative filtering algorithms can be tailored to groups of users rather than individuals. We choose to use only users' positive votes and to present them to the collaborative filtering engine as if they all came from a single user, called a pseudo-user by O'Conner et al. The potential disadvantage of using a pseudo-user approach, that one user may have far more votes than another and hence dominate the recommendations, is avoided in our case as only movies that both users were positive about are used as input to the collaborative filtering.

Help

Because of its initial intended use within demos the VDM had no help features. We will address the addition of help functionality in the next section.

Sounds

The VDM uses sound as well to give a sense of pace and to foster conversation. Stills produce a click when they leave the screen, and sounds are used to differentiate two yes votes, two no votes, and disagreement.

Implementation Detail

The VDM was implemented in C# using a beta of the forthcoming managed code extensions to Direct X.

The Remote controls used for the VDM were Sony RM-VL900s. These were capable of learning the key-up signal from an IR keyboard and so the VDM saw the input from the remotes as a sequence of keystrokes.

CAFÉ INSTALLATION

Having received positive feedback from users during demos of the VDM we were keen to try it out in a public setting. We wanted to understand how the VDM would serve real users as a movie recommendation system, to get users' feedback, and to answer the specific research questions: are the features of sessions (time taken, etc) different depending on the number of users involved?

Figure 2: VDM Cafe Installation

Description

We choose a spot in a corporate campus café close to a coffee bar. The spot was chosen as it was close to our offices so that we could provide maintenance support and because it had over 500 customers every day. Access to the café was restricted by card reader so most café customers were employees but there were also family, friends, academic, and corporate visitors. Customers often use the café with small groups of friends.

Several aspects of the VDM required change before we could install it publicly.

The VDM UI had some features tailored specifically to two users while we now wanted it to work with one to four users. We removed those features and replaced them with more generic counterparts. In particular the tallies of yes and no votes, the agreement indicator were replaced by an enlarged history list in which stills retained their vote tabs so that users could scan the line of recent votes to see how much agreement and disagreement there was, or if one user always voted yes, etc.

We also added help and ease of use features. Help came in the form of a small number of slides explaining the UI and the interaction required. These slides were played across the bottom of the screen when the VDM entered attractor mode or when any user pressed the help button. The remotes were changed so that the buttons were color coded, labeled and any unused buttons were removed. This left us with five buttons per remote: yes, no, skip, help, recommendation.

One problem we had encountered when demoing the VDM was that users had no clear sense of when to stop. So for our installation we demoted the ongoing recommendation to a small DVD thumbnail in the corner of the screen and encouraged users to end the session and get their recommendations by pressing the recommendations button. This moved to a new screen which used all the yes votes gleaned from each user to provide a tailored recommendation and also combined their yes votes to provide five group recommendations – again using a pseudo-user. Although users could take this as the end of a session they were free to carry on. Voting yes at the summary screen caused the VDM to return to the stream of movie stills without saving the users' votes; voting no caused it to return keeping the users' votes.

In order to derive a notion of session from use we augmented the remote controls with capacitance sensors so that the VDM could record when the remotes were in the hands of users. Though we recommend this approach to other researchers it failed to produce usable results in our case due to technical difficulties in the software managing the data gathering.

To get data from the installation we augmented the VDM to record user actions into a SQL database, put up an online feedback form, and provided paper feedback forms near the remotes.

You can see the VDM café installation in Figure 2.

Installation Results

Our notion of session is a derived one. We consider a session started when a vote is received and finished when no votes are received for more than 2 minutes. The choice of a 2 minute timeout is arbitrary and so we additionally examined the data with sessions derived from timeouts of 30 seconds to 5 minutes in 30 second intervals. The data from these other sessions led to equivalent results.

Session length is reported in Figure 3. The average length of sessions goes up as there are more people involved: 166 seconds for 1-user sessions; 385 seconds for 2-user sessions; 690 seconds for 3-user sessions; and 896 seconds for 4-user sessions. This does not represent a drop in efficiency – if we examine the average time taken to first ask the VDM for a recommendation it is 18 seconds for 1-user sessions; 24 seconds for 2-user sessions; 20 seconds for 3-user sessions; and 17 seconds for 4-user sessions. So the increase should be understood in terms of greater scope for enjoyment and conversation.

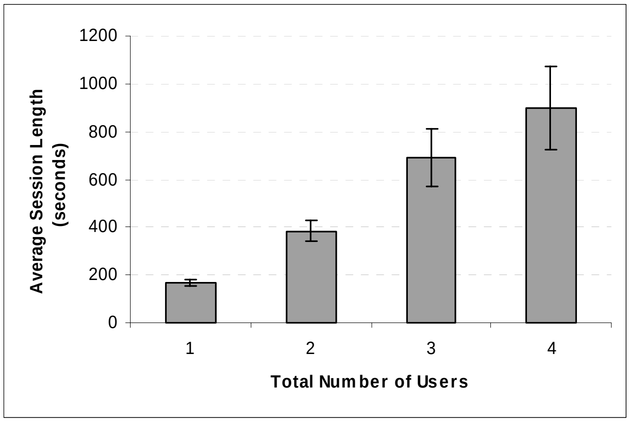

Figure 3: Average Session Length by Total Number of Users. Error bars show standard errors.

The numbers of sessions of different sizes are shown in Figure 4. The figures are 282 1-user sessions, 97 2-user sessions, 20 3-user sessions and 17 4-user sessions. This drop off reflects the ease with which single user sessions can be undertaken as opposed to organizing friends to join you.

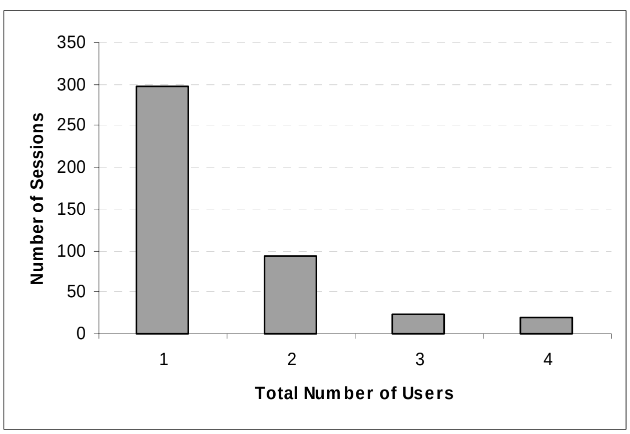

Figure 4: Total Number of Sessions of Different Sizes

In Figure 5 we see the average number of votes per user per second for the varying session sizes. The figures are 0.19 votes per second during 1-user sessions; 0.09 votes per user per second during 2-user sessions; 0.11 votes per user per second during 3-user sessions; and 0.07 votes per user per second during 4-user sessions. As expected, we see that the single user sessions are more focused on voting, while the multi-user sessions include non-voting activity like conversation about the movies. The increased number of votes during 3 user sessions arises from a few long sessions where the skip vote was used more frequently.

This requires further examination. Figure 6 shows the average number of votes per second for differing session lengths. We have grouped the votes into Skip votes and Yes or No votes. The figures are 0.09 Skips per second and 0.13 Yes or No's per second for sessions that last less than 5 minutes (n = 278); 0.06 Skips per second and 0.09 Yes or No's per second for sessions that last between 5 and 10 minutes (n = 57); and 0.11 Skips per second and 0.09 Yes or No's per second for sessions that last more than 10 minutes (n = 59). So skipping a film is less popular than expressing an opinion on it in sessions that last less than 10 minutes. For the longer sessions, skipping is the more common vote.

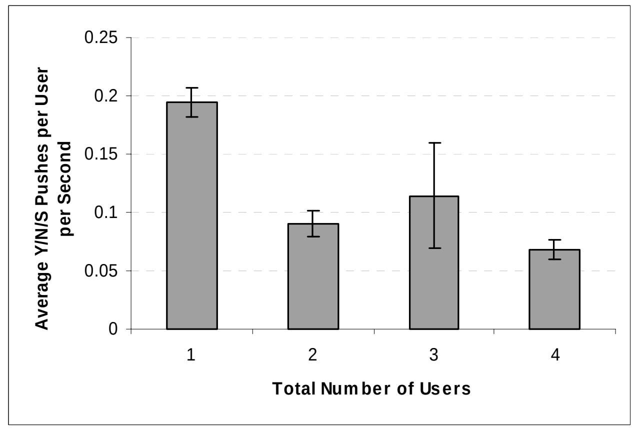

Figure 5: Average Votes for Different Session Sizes. Error bars show standard errors.

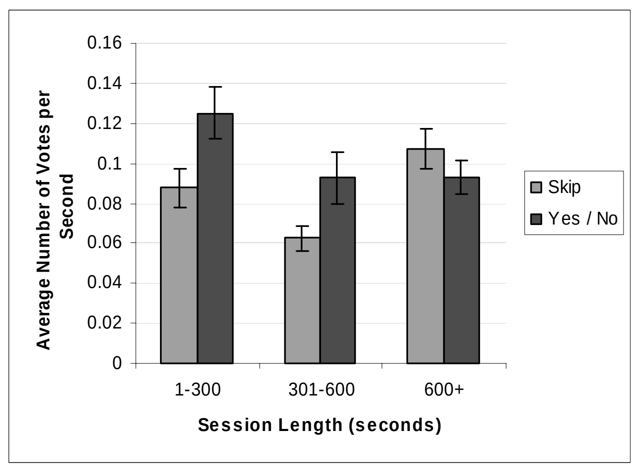

Figure 6: Average Votes for Differing Session Lengths. Error bars show standard errors.

Our explanation for this is that the longer sessions are the result of a more serious intent. The VDM was conceived as an application whose sessions would fill a short period of time. It would be a 'coffee sized task' that users could enjoy during a coffee break. But some users spend far longer than that (our longest session lasts over 40 minutes). Users embarking on these longer sessions are interested in amassing enough precise preference data to obtain highly accurate recommendations. Therefore they do not express an opinion on a movie unless they are sure the opinion is correct – hence they are more likely to skip movies. This is more like the usage of other movie recommendation systems. For example over 7000 MovieLens users have rated over 200 movies each [11]. For the VDM such usage may prove disappointing for two reasons. Firstly we do not require users to log-in and hence cannot reuse their preference data in later sessions. Secondly we were restricted by the number of movie stills we could obtain, so that there were only 823 movies to rate as opposed to the over 4000 used in MovieLens [11]. Our transitory sessions and limited number of movies was OK for the shorter length sessions we had designed for.

The user feedback we received from our web based and our paper based feedback forms revealed general enjoyment of the system but two areas of concern. One was the accuracy of the recommendations and another was speed. We decided to address both of these issues in an experimental setting covered in the next section.

To summarize our discussion of our quantitative results from the café installation we found that there were differences between the sessions with just one user and the sessions with more than one user. We had more single user sessions but they did not last as long. This is not a productivity issue as we saw no marked difference in the time taken to ask for the first recommendation. The single user sessions involved a greater density of votes. This reflects their purpose. Users would use the VDM alone either just to try it out, or to gain movie recommendations. Groups of users had the added incentive of exploring their friends' tastes in movies. This exploration took time and conversation, as we had hoped in our design.

We also observed a redefinition of space as a result of the installation of the VDM. The group of easy chairs near the coffee bar that we installed the VDM amongst changed. Whereas people were often seen drinking coffee and chatting there it became a place for people who wanted to use the VDM. This was not an absolute; people not using the VDM would still sit there, but less so. This was not a desired effect. Solutions we considered include:

- Changing the layout so that the VDM screen was not the focal point of the cluster of seats,
- Install lots of VDMs so that none of them stood out
- Let people become accustomed to the installation

We decided on the last of these approaches, and it was successful. After 3 months a series of observations showed that the space around the VDM was used regardless of whether the VDM was turned on or off.

Another interesting observation from our user feedback was that privacy was not an issue. Several people had commented before the installations that we would get more fidelity in users' votes if the other users could not see how they were voting. Other systems like PolyLens [10] and MusicFX [5] make sure such data is kept as confidential as possible. Our installation exposed the user preference data by definition – its disclosure was one of our key ways of fostering conversation and hence making it enjoyable. So for leisure based applications to be used amongst friends privacy is not a big user concern.

USER STUDY

Purpose

There remained three research questions that required a formal usability study:

1 Do the UI principles we choose affect users' perceptions of the quality of the movie recommendations?

2 Do users prefer our choices on pacing and shared focus over alternatives?

3 Did our UI choices help to foster conversation?

Description

Participants

34 participants (17 female and 17 male) from the greater Puget Sound area were recruited from our company's usability database to participate in the study. Since the VDM was designed for friends to use together, having recruited one participant we asked them to recommend a friend for us to recruit to join them in the study. The primary relationship between the subjects was: four married couples, one pair of colleagues, ten pairs of friends, one pair of siblings, and one pair of housemates. Their occupations were wide-ranging (teacher, accountant, retired, stay at home mom, student, real estate agent, etc). The average length of their relationships was 11 years and 9 months with range 3 months to 51 years. The average age of those participants who declared their age was 37.64 with range 20 to 68 years old. The participants were screened to be intermediate to expert Windows and Internet users, as per validated internal screening tools.

Task & Design

We had paired subjects use the VDM three times, for three minutes each time, under three varied conditions: the standard VDM from the café installation, a faster paced version, and a version with individual (as opposed to shared) focus. Participants completed a brief questionnaire after each of the three sessions and a longer one at the end. Before each use the participants had a short practice session to familiarize themselves with the VDM and any changes since their previous condition. The database of movies was split into four disjoint sets: three sets for the experimental conditions and one for the practice sessions. The ordering of the conditions and of the sets of movie stills was fully counter balanced across the pairs of participants.

The two conditions that varied from the café installation were achieved as follows. The faster paced VDM replaced the 60 second wait period that can elapse once a user has voted while the system leaves the selected still available for the other user to vote on with a 1 second wait. Dropping the shared focus required greater change. We implemented a networked version of the VDM that ran on two PCs with their screens placed side-by-side. This gave the two participants their own independent stream of movie stills to vote on. When either user pressed the recommendation button on their remote both instances of the VDM went to the recommendation screen. The five group recommendations were still based on the intersection of the two users' yes votes and were duplicated on each screen.

Experimental Results

We asked our pairs of participants to score the three conditions on a number of aspects. These included the questions:

- If you had to use the VDM again, which version would you use?
- During which session did you and the other participant have the most conversation?
- Which session had the best recommendations to you both?

The participants could each assign one vote to one of the conditions, giving a combined minimum vote of zero and a combined maximum vote of two from each pair of participants.

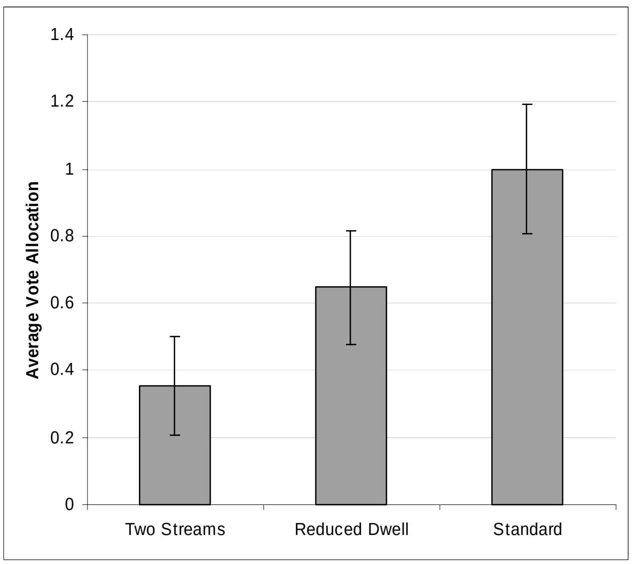

Figure 7: Preferred Version. Error bars show standard errors.

Figure 7 shows how the pairs voted for which version they would prefer to use again. The mean votes were 0.35 for the two streams condition (SD = 0.61), 0.65 for the reduced dwell condition (SD = 0.70), and 1.00 for the standard condition (SD= 0.79). The increased preference for the standard condition over the two streams was significant (post-hoc LSD Least Significant Difference test $p < 0.05$), while the other differences are not significant.

Figure 8 shows how the pairs voted for which version fostered the most conversation. The mean votes were 0.06 for the two streams condition (SD = 0.24), 0.71 for the reduced dwell condition (SD = 0.77), and 1.18 for the standard condition (SD= 0.81). The increased preference for the standard and the reduced dwell condition over the two streams condition was significant (post-hoc LSD test $p < 0.01$ for each), while the difference between the standard and the reduced dwell conditions is not significant.

Figure 9 shows how the pairs voted for which version gave the best group recommendations. The mean votes were 0.41 for the two streams condition (SD = 0.62), 0.65 for the reduced dwell condition (SD = 0.79), and 0.94 for the standard condition (SD= 0.66). The increased preference for the standard condition over the two streams was significant (post-hoc LSD test $p < 0.05$), while the other differences are not significant.

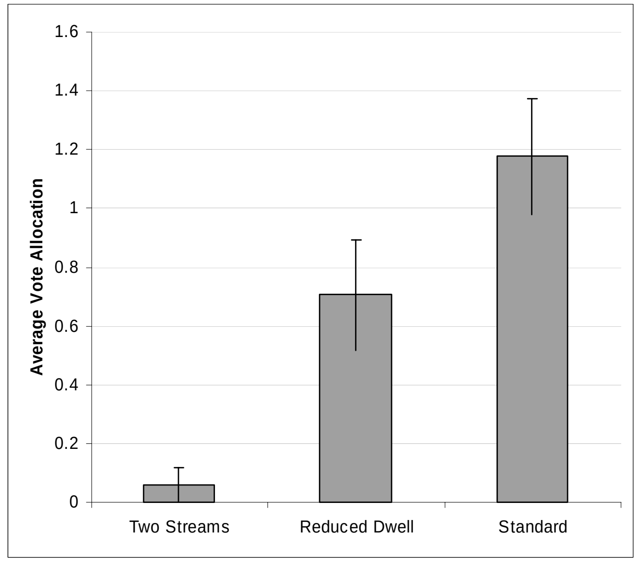

Figure 8: Most Conversation. Error bars show standard errors.

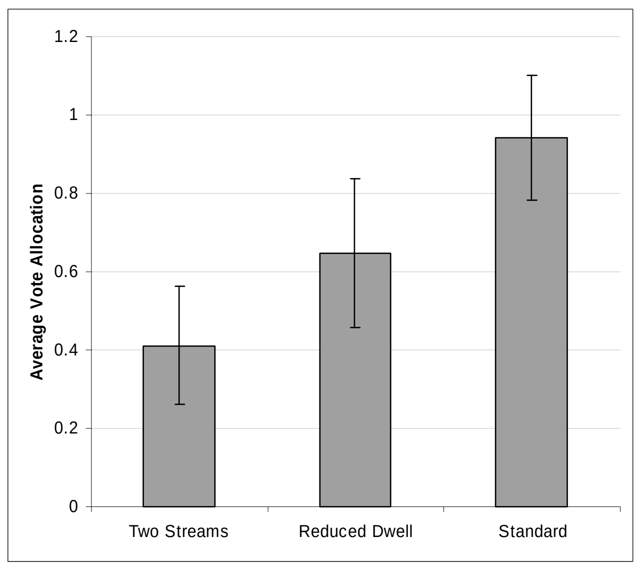

Figure 9: Group Recommendation Quality. Error bars show standard errors.

Discussion of Experimental Results

The results show how our design goals had the intended effect – and an unexpected but important additional effect. Although the sets of movies used were balanced across the participants and the experimental conditions, participants perceived an increase in the quality of the group recommendations from the two streams condition to the standard condition. Sometimes the absolute quality of a recommendation is what is important, but more often the important thing is the perceived quality of the recommendation. Our result shows that by increasing the awareness of each user of the other user's presence through UI enhancements, we also increase their perception of the recommendations quality.

One might explain this in terms of enjoyment, reasoning that tasks are often more enjoyable when done with others, and that the increased enjoyment leads to increased recommendation scoring. However we did not see a significant increase in the perceived quality of the individual recommendations, only the group ones. Our explanation is that whilst a user knows very well when a recommendation aimed at them is inaccurate. But when in a group, the user cannot be sure how each recommendation balances the preferences of the group. Hence the more aware the user is of the other group members' presence the more forgiving the user is of the recommendation.

We also found that our desired effects – increased enjoyment and increased conversation were accomplished by our choice of a shared focus for the co-located users and by giving them sufficient time to take-in and respond to the other user's interactions.

CONCLUSIONS

We have shown how a recommendation system can be designed for a group of co-located users to obtain movie recommendations together. We made our design choices motivated by shared focus, dynamic pacing, and encouraging conversations and have shown how these elements were achieved and their effects on users' enjoyment of the system.

In particular, a shared focus combined with a pace that gave users time to comprehend and comment on other users choices made our system more enjoyable, prompted more conversation between users, and most importantly led to a perceived increased accuracy of the recommendations made.

These results should provide a framework for designers of kiosks and other media applications designed for multiple co-located users. Our results on the increased perception of recommendation quality should be valuable for anyone implementing recommendation systems that may be used by groups of people together, be they friends shopping on the internet, friends renting a movie, etc.

NEXT STEPS

Our next steps are to look at the process of group visual decision making in other leisure settings. In our studies users have come up with an array of situations where they felt that a system like ours would prove useful. These include: restaurants, travel planning, career counseling, interior decorating, hotels, cars, menus, surveys on families, etc.

We also plan to examine how such applications for co-located users might support business meetings and work based group decision making.

REFERENCES

1. Breese, J. S., Heckerman, D., and Kadie, C. (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of Fourteenth Conference on Uncertainty" in Artificial Intelligence, Madison, Wisconsin. Morgan Kaufmann
2. Crossen A., Budzik J., and Hammond K. (2002). Flytrap: Intelligent Group Music Recommendation. *Proceedings of IUI2002*.
3. Gottlieb, H. (2002), "The Jack Principles of the Interactive Conversation Interface", Jellyvision Inc, http://www.jellyvision.com/deal/jackprinc.htm
4. Inkpen K., Ho-Ching W., Kuederle O., Scott S., and Shoemaker B.D. (1999). "This is fun! We're all best friends and we're all playing.": Supporting children's synchronous collaboration. *Proceedings of CSCL 99*.
5. McCarthy J.F. and Anagnost T.D. (1998). MUSICFX: An Arbiter of Group Prefences for Computer Supported Collaborative Workouts. *Proceedings of CSCW98*.
6. Microsoft (2001). Raw Input API. http://msdn.microsoft.com/library/en-us/winui/winui/windowsuserinterface/userinput/rawinput.asp
7. Microsoft and Blitz Games (2001). Fusion Frenzy. http://www.xbox.com/fuzionfrenzy/
8. Microsoft and Bluetorch (2001). Amped, http://www.ampedfreestyle.com/
9. Microsoft and Digital Illusions (2002). RalliSport Challenge. http://www.xbox.com/rallisportchallenge/
10. O'Conner M., Cosley D., Konstan J.A., and Riedl J. (2001). PolyLens: A Recommender System for Groups of Users. *Proceedings of ECSCW2001*.
11. Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., Riedl, J. (2002), "Getting to know you: learning new user preferences in recommender systems" in 2002 International Conference on Intelligent User Interfaces, ACM
12. Stewart J., Raybourn E.M., Bederson B., and Druin A. (1998). When Two Hands Are Better Than One: Enhancing Collaboration Using Single Display Groupware. *Proceedings of CHI 98.*
13. Two-Way TV. http://www.twowaytv.co.uk/