Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/preface.pdf

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces – Preface

Bart P. Knijnenburg, Dirk Bollen
Human-Technology Interaction group
Eindhoven University of technology, The Netherlands
{B.P.Knijnenburg,D.G.F.M.Bollen}@tue.nl

Lars Schmidt-thieme
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

1. INTRODUCTION

In his keynote speech at the 2009 RecSys conference, Francisco Martin indicated that the main challenge in recommender system industry is not to discover algorithms that provide good recommendations, but to provide users with a usable and intuitive interface for presenting these recommendations and eliciting feedback.

Unfortunately, the research on 'Human-Recommender Interaction' is scarce. While algorithm optimization and off-line testing using measures like RMSE are standard procedure in the RecSys community, theorizing about consumer decision processes and measuring user satisfaction in online tests is much less common.

Meanwhile, researchers in Marketing and Decision-Making have been investigating consumer choice processes in great detail, but only sparingly put this knowledge to use in technological applications. Likewise, the field of Human-Computer Interaction has been studying the usability of interfaces for ages, but does not seem to connect the dots between research on consumer choice, and recommender system interfaces.

The UCERSTI workshop tries to bridge the gaps between recommender systems, human computer interaction and marketing/decision-making research by providing a platform for Human-Recommender Interaction research.

2. INCLUDED PAPERS

These workshop proceedings include the following papers:

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

Visual Nudges for Enhancing the Use and Produce of Reputation Information

Kristiina Karvonen[1], Sanna Shibasaki[1], Sofia Nunes[1], Puneet Kaur[1], Olli Immonen[2]

[1]Helsinki Institute for Information Technology HIIT
P.O.Box 19800 Aalto
FIN-00076 +358 9 470 28362

{kristiina.karvonen, sanna.shibasaki, sofia.nunes, puneet.kaur@hiit.fi}

[2]Nokia
P.O.Box 407
00045 Nokia Group, Finland +358 71 800 8000

olli.immonen@nokia.com

ABSTRACT

In this paper, we aim to analyse the current level of usability on ten popular online websites utilising some kind of reputation system. The conducted heuristic and expert evaluations reveal a number of deficiencies on the overall usability of these websites, but especially on how the reputation information is currently presented. The low level of usability has direct consequences on how accessible and understandable the reputation information is to the user. We also conducted user studies, consisting of test tasks and interviews, on two websites utilising reputation information. The results suggest why the currently provided information remains under-utilised and, to a great extent, goes undetected or gets misinterpreted. On basis of the work so far, we propose ways to overcome some of the current problems by changing, rearranging and grouping of the visual elements and visual layout of the reputation information offered on the sites. The enhanced visualisations create "visual nudges" by enhancing the key elements in order to make users notice and use the information available for better and more informed decisions. .

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces: Evaluation/Methodology

General Terms

Design, Security, Human Factors

Keywords

Usability, heuristics, expert evaluation, user study, recommendation, reputation, visual nudge, user interface design

1. INTRODUCTION

As Internet services and peer-to-peer systems currently are lacking in the traditional indicators of trustworthiness [3], being able to differentiate between a good offer and a bad one in an easy manner is not trivial. In the peer-to-peer markets especially, information about the reputation of the various parties in the online transactions – the buyer, seller, and venue – can help to make good decisions and diminish the risks involved [5].

Reputation systems have grown into a prominent means to gather and provide such information about the quality of the offering and its seller for the end user. A reputation system operates by computing reputation scores for some set of objects, such as services or items on sale, within a certain community or domain. The scores can typically be computed on basis of a collection of opinions – usually ratings – that other entities hold about the objects, by employing a reputation algorithm to calculate reputation scores based on the received ratings, which are then published. Reputation information typically represents users' opinions about a particular product, service or peers [5].

Reputation information can be textual (e.g. descriptions, reviews) or visual (e.g. images, symbols, statistical visualisations), or, usually, a combination of the two. However, currently the reputation information is often presented in such a way that may make it hard to notice and to interpret. To make things worse, according to our heuristic and expert evaluations, the overall level of usability on the sites offering reputation information is often bad enough to stop users from effectively having the reputation information at their disposal, as it goes undetected: if the user cannot find the functionality, the functionality is not really there [12]. The reputation information is not utilised as guidance in the way it could and should be.

Which parts of the reputation information is presented visually needs to be carefully selected: Our user studies [9][16] evaluating websites that use reputation systems have shown that the visually prominent parts of the reputation information offered gets center stage, regardless of its actual usefulness and relevance for the decision making. Furthermore, cohesion between the various reputation elements is often missing and the reputation information is experienced as scattered, with unrelated pieces of information that are being used in random combinations that is dictated by their visual prominence, rather than by their actual importance for the decision-making.

To further investigate the described issues we have evaluated ten more websites of different categories (news, shopping, social networking etc.) that employ some kind of reputation system. The main objective of the usability evaluations was to evaluate the current level of usability of these services, and how well the standard set of heuristics from Nielsen [13] works for sites with reputation information, or if they need additional rules of thumb. In the expert evaluations, we were focusing on the reputation information and how it is visualised in order to understand what works, what fails and how things could be improved.

As the visual prominence seems key for better utilisation of the reputation information, we introduce the idea of *visual nudging* for improving the usage and production of reputation

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

information to enable better and more informed decision-making. "Nudging", a term introduced by Thaler et al as a way to enhance decision-making [19], in this context means that by enhancing the key elements of the reputation information that the user *should* be looking at in order to reach a good decision, we aim to gently influence the users' behavior by focusing their attention in relevant direction. The visually prominent elements are intended to serve as nudges. A nudge can alter the users' behavior in a predictable way without forbidding any options or significantly changing their economic incentives [19]. As indicated by our previous studies [9], nudging through the visual means could be most effective as visual elements are gaining the users' attention. Further, better visualisation may also help to create more interest in contributing to the reputation information (commenting and rating), as currently the ratio between all users of a site and those who actually actively add to the reputation information is often quite low [add ref or take out].

We will first present the background for the current study, the previously conducted user studies together with the earlier work done in this area. We will then proceed with the usability evaluations for the additional websites and discuss the findings. We will conclude by summarising the lessons learned on what kind of usability issues we currently see as most pressing on the websites utilising reputation systems, and how they could be improved on, especially focusing on the key role of the visual elements and their prominence for the overall usability of such websites.

2. BACKGROUND

Reputation information is typically presented by both visual and textual means.

2.1 Visual reputation information

Currently, the most common way to present *visual* reputation information is to use star symbols to represent the current rating of the item under scrutiny (Figure 1). Other symbolic icons commonly used for visual reputation information include "thumbs up" or "thumbs down" and a scale consisting of circle symbols (Figure 2).

Most common representations of reputation information are used to communicate the popularity rate of the product or service based on users' votes. Usually, the user is able to see the amount of votes given describing the popularity or how much the product is "liked". However, this information is not revealing the *scale* of the information, and the user may be left with confusion: *What is the difference between three or four stars? How many stars a good product usually gets? How many ratings can be considered "a lot of ratings" in this service?* Because of this ambiguity, the quality of the reputation information is experienced as questionable: *What do the ratings actually mean (to me)? How credible are the ratings? How are the ratings calculated?* For the users, the transparency of the information [17][18] is missing.

Figure 1. Examples of usage of the star symbols as reputation visualisation in some popular websites

Figure 2. Example of other commonly used symbolic icons for reputation information

2.2 Textual reputation information

Possibly, partly due to all of these problems in the visually presented reputation information, the textual information is currently considered more important for the users: Reliance on peer reviews has become everyday news. For example, USAToday has recently reported the growing importance of peer reviews, stating that "customers are increasingly vocalising their experiences online for other travelers to read" [22]. In another article, online ratings and reviews were considered almost twice as significant as brand and reputation when choosing a hotel [21].

Online reviews have indeed become increasingly popular as a way to judge the quality of various products and services [4][8][11]. Even when popular and used, the textual reputation information has its own troubles. The basic usability problems related to how the information is presented hinder the efficient use of the reviews. The user is encountering a burden of finding the relevant information out of sometimes an excessive amount of textual feedback. Furthermore, in a recent study by Jurca et al [8], the reviewing behavior can also include a variety of biases.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

2.3 Trust and risk

In the context of downloading, trust and risk perception also become an issue. For the online user, the perceived credibility of a website or a service has a strong impact on the trust level and risk perception [5]. As it has been studied before [1], visual or aesthetic factors are linked to a website's credibility – a good first impression, strongly based on the visual representation, can set the trust level towards the service in a matter of milliseconds [10]. Investing on a visually pleasing user interface (UI) has been found to enhance a positive user experience of web pages [7][14].

3. EARLIER WORK

In our earlier work [9][16], we have studied the basis of the actual usage, usability and the ways of utilisation of the reputation information in the context of websites that offer mobile applications for downloading. Our studies focused on two websites; 1) WidSets, which was a website for downloading and developing mobile applications ("widgets"), launched in October 2006 by Nokia (www.widsets.com) and 2) Nokia Ovi Store (www.ovi.com), Nokia's Internet service offering services in various areas such as games, maps, music, and mobile applications. Ovi replaced the WidSets site in April 2009. Our study on Ovi focused on the part of the service offering downloadable mobile applications.

In the study for the WidSets website [9], we were focusing on the current usage of the reputation elements on the website. The results indicated that the *visually prominent UI elements* of the site acted as the main sources of information when making decisions about downloading widgets, while less prominent information was, for the most, overlooked. Therefore, we were able to conclude that any information that is de facto important for the decision making should also be presented as visually prominent in order to gain the users' attention. The question of whether the elements should be *presented as an aggregation of the different elements or separately*, allowing users to utilise the information in a more independent fashion, could not be determined on basis of the studies and thus became one of the questions to be resolved by further studies.

As a direct continuation of the WidSets study, we conducted another study focusing on Ovi and how the online reputation information currently offered in Ovi is understood and utilised by its users [16].

Our results again showed that the reputation information available was not efficiently utilised. According to our interpretation, the *lack of cohesion* between the reputation elements hinders the understandability and use of the information available. Users also reported that they found the credibility and quality of the reputation information to be questionable, which may be the result of the inconsistent and ambiguous way of presenting the information. Users were currently not able to find the relevant information and thus also not able to form an overall view or an understanding about the content and the message of the reputation information.

Based on the results from these studies we suggested [16] that in order to help users making full use of the reputation information, a *visually prominent aggregation of the various reputation elements* would be helpful. According to our studies, the users also preferred the decision making process to be "quick and easy". Answering these demands requires efficient composition of information from different sources. As humans are experts in processing visual information, presenting the information visually, in graphical form is also likely to ease and enhance the information processing.

4. RESEARCH QUESTIONS AND METHODOLOGY

The previous studies showed that there is a lack of visual prominence and cohesion between the different reputation elements, and the reputation information was under-utilised. The findings led to the formulation of the following **hypotheses**:

- The websites offering reputation information had problems with usability;
- More specifically, the reputation information provided has bad usability;
- Visual prominence of the reputation elements is guiding the decision-making process on these sites;
- The visually prominent elements on the websites are "wrong";
- Visual nudging is not working on the websites to enhance the decision-making process.

The basic **research question** behind the study is: *"Why is the reputation information underutilised?"* By addressing this research question, and armed with an initial understanding about the importance of the visual elements, we aimed at analysing how the reputation information is currently displayed across the selected sites.

Among the various **methods** available in the field of Human Computer Interaction (HCI), *heuristic evaluation* based on Nielsen's heuristics [12] was chosen as the basic method to analyse the sites offering reputation information. The heuristic evaluation was complemented with *expert evaluation* focusing on the visual elements of the sites.

Heuristic evaluation is a form of usability inspection where usability specialists or other evaluators judge how the object of study, e.g. a website, passes on an itemised list of established usability heuristics [12][15]. Preferably, the evaluators are experts in human factors or HCI, but less experienced evaluators can also follow the heuristics checklist and produce a report of valid problems. Expert evaluation is a more free-form analysis of a given object under observation, based on the expert's experience, often focusing on certain elements of the object [2].

With the evaluations, we aimed at gaining an understanding of the usability issues and to potentially formulate additional heuristics for reputation information.

5. THE STUDY

The websites chosen for the usability evaluation were well-known sites, and selected on basis of their general popularity[1]:

[1] http://www.google.com/adplanner/static/top1000/#, http://www.alexa.com/topsites,

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

- Amazon (shopping), www.amazon.com
- eBay (shopping), www.ebay.com
- TripAdvisor (hotel and vacation reviews), www.tripadvisor.com
- LinkedIn (networking tool), www.linkedin.com
- YouTube (video sharing), www.youtube.com
- Yelp (reviews and recommendations for local businesses), www.yelp.com
- Digg (social news website), digg.com
- IMDb (movie and serial reviews), www.imdb.com
- NowPublic (social news website), www.nowpublic.com
- AppStore (Apple's store for iPhone applications). www.apple.com/iphone/apps-for-iphone/

The evaluations were performed by four evaluators: one senior HCI expert (> 10 years of experience), 2 expert (>2 years of experience) and one non-expert (< 1 year of experience). The expert evaluation focused on how the reputation information was presented on the selected sites.

6. ANALYSIS OF THE USABILITY EVALUATIONS

Table 1 summarises the outcomes of the usability evaluations against Nielsen's heuristics. We will now present the findings of the expert evaluations on the reputation information website by website, focusing on the main findings. The findings are marked either with (negative) or (positive).

Amazon

The different pieces of information are presented similarly, as if having the same value (e.g. product details and important information). This makes retrieving information for the decision-making a hard task. (Figure 3).

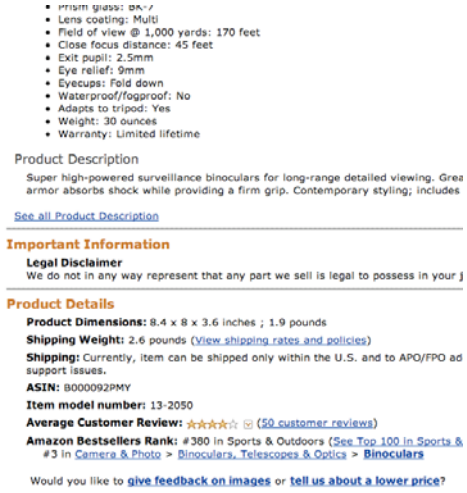

Figure 3. Different types of information similarly presented

The website presents the rating's information through a chart with detailed information about how many users rated the item and how, as well as a direct access to their reviews.

Information about the seller is presented clearly.

Users can access the list of top reviewers, i.e. the ones with the most useful reviews.

eBay

Information about the overall purpose of the website is hard to find even when registering (statement of purpose).

The user cannot sort other users' reviews about a seller by any other category except "date", the default category. In case a seller has both positive and negative reviews, the user will have to scroll through all the reviews to find the negative ones. This might be very time-consuming (Figure 4).

Both the ratings about the seller and the way the feedback is calculated are clearly presented to the user.

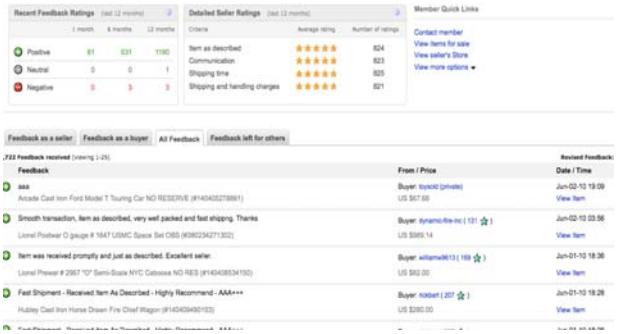

Figure 4. Sort reviews

TripAdvisor

The visualisation of the rating system is ambiguous. A novice user might be confused by the two different ways of showing the ratings 1) thumbs and 2) circles. The actual meaning of the symbols becomes clear only by the time the user writes a review: thumbs are associated with a separate question - "would you recommend this to a friend?" (Figure 5); circles represent the rating.

Figure 5. Confusing information

The number of reviews is not consistent. The addition of all the ratings provides a number, which is different than the one presented along with the written reviews and still different from the one obtained when the user clicks the "clear filters" option. This might jeopardise trust in the reputation system.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

		Amazon	eBay	TripAdvisor	Linkedin	YouTube	Yelp	Digg	Imdb	NowPublic	AppStore
1.	Visibility of system status	X	X	√		√	√	√	X	√	√
2.	Match between system and the real world	√									√
3.	User control and freedom	√ / X	X		√	X	X	X	√ / X	X	X
4.	Consistency and standards	X		X	√ / X	X	X	X	X		X
5.	Error prevention	√	√	√	√	X		√		√ / X	√
6.	Recognition rather than recall	X	X	√	√	√	√ / X	X			√ / X
7.	Flexibility and efficiency of use		√	√	√	√	√	√	√	√	√
8.	Aesthetic and minimalistic design	X	X	X	√			X	X	X	√
9.	Help users recognize, diagnose, and recover from error										
10.	Help and documentation	√	X							√	

Table 1. Overall outcomes of the heuristic evaluation. The symbol √ was used when there were more good aspects than problems, the X was used when the problems were more than the good aspects and the √ / X symbols when the number of problems and good aspects was balanced

Information provided is not clear. For example the rating information provided for hotels consists of three different ratings (Figure 6).

The different elements of information are presented as having the same value, and without a clear structure to guide the user, which makes retrieving information a time consuming task.

The target of the reputation and the reputation elements were not easily distinguishable.

While reading the reviews, the user can see the reviewer profile with just a mouse hover, which provides an easy access to the information, prevents the disruption of the task and adds quality to the user experience.

Figure 6. Confusing rating information

LinkedIn

The UI does not provide a clear guidance of what are the goals of the website, how it should be used and what is the order of importance of the content. This information is hidden behind an unnoticeable link, which makes it hard for the novice user to detect.

The users' own recommendations are listed, enabling comparison between recommendations, and adding transparency to the system.

YouTube

After having rated a video as negative or positive, the user is not allowed to undo the action. This adds unreliability to the system especially as it is possible to click on the rating accidentally.

User is not allowed to delete a video previously rated as "Liked" from the "liked videos" view (Figure 7). The only actions allowed are adding it to a playlist or to a list of favorites. In order to delete a video previously rated as "liked" the user has to perform too many steps. First, the user has to open the "liked videos" view, add the selected video to a playlist or to favorites and only then remove the video. This is time consuming and counter intuitive as the user has to perform a contradictory operation – “add to favorites” - to the one they actually intend to perform.

The system does not provide a confirmation or an option to undo the action of reporting another user. This might generate

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

unreliability in the reputation information as users can report and be reported by accident.

There is specific statistical information about the history, popularity and spread of the videos, which contributes to the transparency of the website.

Information provided under "views" shows a detailed pictorial and statistical representation of activity frequency over time and per location.

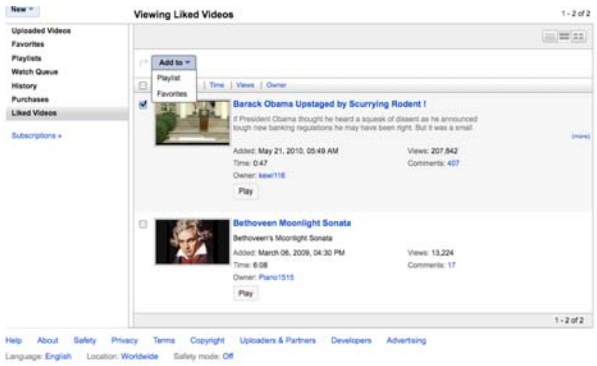

Figure 7. No delete option

Yelp

The users have access to the amount of reviews for a specific place but cannot see the relationship between other reviewed places. Even if all the reviews are positive and the place has a certain number of stars it does not provide information about its quality when compared to other places in the same area.

After rating a review as useful, funny or cool, the user is provided with feedback and the number of ratings is immediately updated, which evokes reliability in the system.

The system provides the option to undo the ratings to other users' reviews, which allows the user to correct potential mistakes and adds more trustworthiness to the ratings.

The website provides a graphical and clear explanation of ratings and ratings over time. It clearly details how the overall ratings are obtained.

The basic review contains plenty of information about the reviewers' reputation, making the relevant information immediately available to the user and the reputation of the review itself can also be seen.

By presenting diverse information about the reviewed target and the reviewer community on the first page the website guides the novice users and keeps their interest in exploring the website.

Digg

The main page does not provide information about what is "Digg" or how it works. The lack of directions might make the novice user confused about the purpose of the website.

Advertisements were presented as having the same value as the information the user was looking for.

The system does not allow the user to delete a previously provided comment.

The scale of the "Top" is ambiguous. The user is not able to distinguish the timeframe of the "tops" and might get confused.

When clicking the icon corresponding to the number of "diggs", the user is directed to a page presenting the comments. This is counter-intuitive since the user expects to see a list related to the number of "diggs", instead of the comments regarding the news. The "how many diggs"- icon is the most prominent element of the page, hence it should provide the expected information.

After digging an article the system provides good feedback and updates the results immediately, which contributes to the overall reliability of the system.

The site enables users to evaluate one another's comments, which might contribute to establish or strengthen the community feeling.

IMDb

If the user rates the same movie more than once the system provides a feedback message saying the vote was counted, which might be misleading.

The user profile, accessed through the username link, only contains a list of the reviews that the user has made. The more informative user profile is accessible through an additional link on the page presenting the users' reviews. This jeopardises the system's consistency.

The reputation information and the links to reputation information are presented among the general information about the movie. The information is mainly presented in the form of text. The first link on the page dedicated to the reviews is blended among the general textual information and the links, which requires an extra effort from the user in order to find relevant information and differentiate between different types of information provided.

User cannot distinguish the relationships between popularity and rating of the movies. The info button on MOVIEmeter (question mark) gives some additional information but does not resolve the issue as the users may have a hard time understanding how the percentages are formed and how to interpret them.

The website provides detailed user ratings, and allows the user to access information about the voting trends for specific categories.

The website uses weighted average for unbiased ratings, which eliminates the ratings that are only intended to change the overall rating in their benefit, adding reliability to the reputation information.

The website also provides links to external reviews, which contributes for the feeling of transparency.

NowPublic

Information elements and advertisements are hard to tear apart. The small boxes of information and advertisements create

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

a cluttered look for the UI and the vertical page structure does not support a natural flow of information retrieval.

The "recommend" icon does not provide clear information about if the user is recommending the other member or their posts. This might affect the results, in case the users do not understand what is recommended (Figure 8).

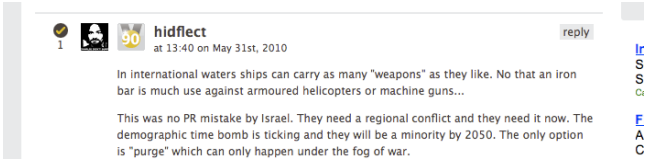

Figure 8. Misleading icon

The website provides a guidance pop-up window for novice users as a starting page, which gives immediate information about the purpose and usage of the website.

The website provides detailed and clear information about getting promotion by points and an explanation about the meaning of the user ranking.

The members are given points according to different categories of posts. This motivates contribution as it might be seen as recognition.

The ranking status of the members, based on their individual points, is presented visually and in a clear way.

AppStore

An option to read more information in the reviews - expand text – is provided, but the user cannot go back to the condensed text, which can make the page cluttered.

The site does not offer access to more details about the star ratings or all customer reviews unless the user uses the iTunes software to view applications.

The user has no information about the way the ratings are formed except for the fact that they are based on the reviews.

The user can easily sort the reviews by several categories that are provided on the left column. This adds efficiency and transparency to the presented information, as the user is able to easily find both positive and negative reviews.

The website provides a list of accessories rated and suggested by staff, which makes it easy for a first time user to navigate through what is available in the store.

When user clicks on a product, all information is provided in three sections – 1) a description with snapshots, 2) ratings and reviews by users and 3) Q&A section, with questions asked and answered by other users. This provides a complete and detailed overview of the products, contributing for transparency.

The website offers visibility for the developer, which may enhance both the willingness to contribute and the trustworthiness of the contributions.

7. DISCUSSION

A general problem found in most of the analysed websites was *a cluttered UI* and the fact that the all available information was presented in a similar fashion as if having the same value, which may cause confusion and mislead the user: The *nudge to look at information that is relevant is missing*. The elements available are presented in a way that does not guide the users' attention to the relevant information while making decisions. Another main problem was related with the *lack of interrelation between the different reputation elements*. This has a negative effect on the information credibility provided by these elements. It may also affect the users' willingness to contribute as it is unclear how the contribution will affect the offering.

On basis of the usability evaluations, the current level of usability on the studied websites has general usability problems that are big enough to jeopardise the use of the sites altogether. Moreover, when it comes to how reputation information is currently offered, the level of usability can be described as remarkably low. Improvements in distinguishing and understanding different types of information available and visual nudges for how they should be utilised by the user in the decision-making process can easily be suggested:

- Clearly *distinguish between distinct sources of information:* the service provider, the reputation system, advertisements, other users and what is actually meaningful – highlight the relevant information and guide the users task-flow;
- *Tie together the different instances of reputation information* to form a coherent set of information where different elements support each other;
- *Promote transparency*: clearly show where the reputation information comes from and how it is formed.

There are also social aspects related to understanding, or accepting the information. The results of our earlier studies and those by others have indicated that reputation information available in textual format, in form of peer reviews in writing, has a big importance in online decision-making [9][8][11][16]. Although the quality of the reviews is sometimes seen as questionable as already discussed, reading peer reviews or comments undeniably is currently the most reported element to be used to make decisions online, when available. However, a closer look may reveal that users may report reviews as the main information source more readily than visual impressions, as users may not be able to reflect on their visual impressions that not only are hard to put into words, are also to a great extent formed automatically and unconsciously [10]. Because of this, users may over-report the importance of the textual information, and under-report the importance of the visual impressions, as they may not be fully aware of it.

Some ways to take all the above-mentioned aspects into account and enhance the utilisation of all reputation elements conjointly is likely to include creating visually prominent, real-time links between the users. When users are exposed to appropriate amount of social data about one another, it tends to increase the activity of giving contributions [6]. The user profiles should also be presented in a visually attractive and motivational way in order to promote participation and contributions [20]. By visual

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper1.pdf

nudges – making the relevant information visually prominent – users can be helped towards more sound and informed decisions in risky online situations.

8. REFERENCES

[1] Alsudani, F. and Casey, M. 2009. The effect of aesthetics on web credibility. In *Proceedings of the 2009 British Computer Society Conference on Human-Computer interaction* (Cambridge, United Kingdom, September 01 - 05, 2009). British Computer Society Conference on Human-Computer Interaction. British Computer Society, Swinton, UK, 512-519.

[2] Baauw, E., Bekker, M. M., and Markopoulos, P. 2006. Assessing the applicability of the structured expert evaluation method (SEEM) for a wider age group. In *Proceedings of the 2006 Conference on interaction Design and Children* (Tampere, Finland, June 07 - 09, 2006). IDC '06. ACM, New York, NY, 73-80

[3] Bhattacharjee, R. and Goel, A. 2005. Avoiding ballot stuffing in eBay-like reputation systems. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems* (Philadelphia, Pennsylvania, USA, August 22-22,2005). P2PECON' 05. ACM, New York, NY, 133-137.

[4] Cheung, M.Y, Luo, C, Sia, C.L, Chen, H. Credibility of Electronic Word-of-Mouth: Informational and Normative Determinants of On-line Consumer Recommendations. *International Journal of Electronic Commerce* 13, 4 (2009) 9-38.

[5] Egger, F. N. Affective Design of e-commerce User Interface: How to Maximize Perceived Trustworthiness? *Proceedings of the International Conference on Affective Human Factors Design.* London: Academic Press (2001), 317-24.

[6] Harper, F. M. The impact of Social Design on User Contributions to Online Communities. Doctoral Thesis. UMI Order Number: AAI3358616.University of Minnesota, 2009.

[7] Hartmann, J., Sutcliffe, A., Angeli, A. D. Towards a Theory of User Judgment of Aesthetics and user Interface Quality. ACM Transactions on Computer-Human Interaction (TOCHI), Article No. 15, ACM New York, NY, USA, 15(4), 2008.

[8] Jurca, R., Garcin, F., Talwar, A., and Faltings, B. 2010. Reporting incentives and biases in online review forums. *ACM Trans. Web* 4, 2 (Apr. 2010), 1-27.

[9] Karvonen, K., Kilinkaridis, T., Immonen, O. *WidSets: A Usability Study of Widget Sharing*, in: T. Gross et al. (Eds.): INTERACT 2009, Part II, LNCS 5727, pp. 461–464, 2009. The Proceedings of INTERACT 2009, 12th IFIP TC13 Conference in Human-Computer Interaction, August 24-28, 2009, Uppsala, Sweden

[10] Lindgaard, G., Fernandes, G., Dudek, C., Brown, J. Attention Web Designers: you have 50 Milliseconds to Make a Good First Impression! *Behavior & Information Technology* 25, 2 (2006), 115-126.

[11] Park, D.H., Lee, J., Han, I. The Effect of On-line Consumer Reviews on Consumer Purchasing Intentions. *International Journal of Electronic Commerce* 11, 4 (2007) 125-148.

[12] Nielsen, J. Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, 1999

[13] Nielsen, J. Usability Engineering. Academic Press, 1993.

[14] Robins, D., Holmes, J. Aesthetics and Credibility in Web site Design. Information Processing and Management: An International Journal, 44(1):386–399, 2008.

[15] Sears, A. Heuristic Walkthroughs: Finding the problems without the noise. International Journal of Human-Computer Interaction, 9(3) (1997) 213-234

[16] Shibasaki, S., Nunes, S., Immonen, O., Karvonen, K.: Understanding Online Reputation Information (unpublished manuscript under submission)

[17] Sinha, R., Swearingen, K. The Role of Transparency in Recommender Systems. *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*. ACM Press (2002).

[18] Suh, B., Chi, E. H., Kittur, A., and Pendleton, B. A. 2008. Lifting the veil: improving accountability and social transparency in Wikipedia with wikidashboard. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1037-1040.

[19] Thaler, R. H., Sunstein, C. R. Nudge: Improving Decisions About Health, Wealth and Happiness. Yale University Press (2008).

[20] Vassileva, J. and Sun, L. 2007. An improved design and a case study of a social visualization encouraging participation in online communities. In: Proceedings of the 13th international Conference on Groupware: Design Implementation, and Use (Bariloche, Argentina, September 16 – 20, 2007). J. M. Haake, S. F., Ochoa, and A. Cechich, Eds. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 72-86.

[21] Ye, Q., Law, R., Gu, B. The Impact of online user reviews on hotel room sales. *International Journal of Hospitality Management* 28, (2009) 180-183., R: Hearing Online Critiques. USA today, 22.3.2010

[22] Yu, R. Hearing Online Critiques. USA Today, 22.3.2010. http://www.usatoday.com/MONEY/usaedition/2010-03-23-businesstravel23_ST_U.htm

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper2.pdf

Recommender Systems: Investigating the Impact of Recommendations on User Choices and Behaviors

Robin Naughton, Xia Lin

The iSchool at Drexel, College of Information Science and Technology
3141 Chestnut Street, Philadelphia, PA 19104 USA
{rnaughton,xlin}@ischool.drexel.edu

ABSTRACT

Recommender systems have been used in many information systems, helping users handle information overload by providing users with a way to receive specific recommendations that fulfill their information seeking needs. Research in this area has been focused on the recommender system algorithms and improving the core technology so that recommendations are robust. However, little research is focused on the user-centered perspective of recommendations provided by recommender systems and the impact of recommendations on user's information behaviors. In this paper, we describe the results of an exploratory survey study on a book recommender system, LibraryThing, and the impact of recommendations on user choices, particularly what users do as a result of getting a recommendation. Based on survey respondents, our results indicate that users prefer member recommendations rather than the algorithm-based automatic recommendations and about two third of users that responded are influenced by the recommendations in their various information activities.

Categories and Subject Descriptors

H5.3 [Information Interfaces and Presentations]: Group and Organization Interfaces *collaborative computing, organizational design, web-based interaction*

General Terms

Computer applications, Design, Evaluation

Keywords

Recommender systems, user-centered design, survey study, user information behaviors

1. INTRODUCTION

Recommender systems offer a solution to the problem of information overload by providing a way for users to receive specific information that fulfill their information needs. These systems help people make choices that will impact their daily lives and according to Resnick and Varian [10], "Recommender Systems assist and augment this natural social process." As more information is produced, the need and growth of recommender systems continue to increase. One can find recommender systems in many domains ranging from movies (MovieLens.org) to books (LibraryThing.com) to e-commerce (Amazon.com). Research into this area is also growing to meet the demand, focusing on the core recommender technology and evaluation of recommender algorithms. However, there's a need for user-centered research into recommender systems that looks beyond the algorithms to people's use of the recommendations and the impact of those recommendations on people's choices. With this in mind, the study objective is to understand the impact of recommendations on user choices and behavior through the use of recommender systems, and this paper presents the results from an exploratory survey of users of a book recommender system, LibraryThing, focusing on whether users follow the recommendations they receive and how those recommendations impact their choices, particularly what users do as a result of getting a recommendation.

2. LITERATURE REVIEW

2.1 Recommender Systems

Resnick and Varian [10] chose to focus on the term "recommender system" rather than "collaborative filtering" because "recommender system" may or may not include collaboration and it may suggest interesting items to users in addition to what should be filtered out. By using the term "recommender system," it becomes clear that the system is not just about the algorithm, but rather the overall goal. It also becomes an umbrella term for different types of recommender systems that uses various algorithms to achieve their goals. Recommender systems can have algorithms that are constraint-based (question and answer conversational method) [3], content-based (CB) (item description comparison method), collaborative filtering (CF) (user ratings and taste similarity method), and hybrid (a combination of different algorithms) [7, 15]. The collaborative filtering technique has gained in popularity over the years [5] and the social networking aspects help to strengthen the filtering techniques. The hybrid technique combines collaborative filtering with content-based techniques to capitalize on the strength of each method.

2.2 Evaluation of Recommender Systems

Research on recommender systems algorithms is very active and seeks to enhance current recommender systems. However, as recommender systems improve, it is important that there is user-centered research on the evaluation of recommender systems. According to Herlocker, et al [5], "To date, there has been no published attempt to synthesize what is known about the evaluation of recommender systems, nor to systematically understand the implications of evaluating recommender systems for different tasks and different contexts." Herlocker, et al [5] focused extensively on the problems of evaluating recommender systems, presenting methods of analysis and experiments that provides a framework for evaluation. Identifying three major challenges, they point out that algorithms perform differently on different datasets, evaluation goals can differ, and deciding on measurement in comparative evaluation can be a challenge [5]. Hernandez del Olmo and Gaudioso [6] proposed an alternative evaluation framework for recommender systems that focuses on the goal of the recommender system. They indicate that there's a

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper2.pdf

shift in the field to a broader and general definition of recommender systems that focuses on guiding users to "useful/interesting objects" [6]. This redefining of the recommender system goals also frames the redefining of the recommender system framework, implying that evaluation can be based on goal achievement of guiding the user and providing useful/interesting items [6]. By dividing recommenders into these subsystems, the authors suggest that each recommender system will have one of the two subsystems more active than the other and the closer they are in terms of activity, the closer they are to achieving the global objective of the recommender system.

The work of Herlocker, et. al [5] and Hernandez del Olmo and Gaudioso [6] offer evaluation frameworks that function across different domains and algorithms. However, they are still steps away from focusing on evaluating recommender systems from the user perspective. A few steps closer is research focused on improving the user experience. Celma and Herrera [2] "Item- and User-centric evaluation" methods to identify novel recommendations based on CF and CB systems, and found that users perceive recommendations through CF are of higher quality "even though CF recommends less novel items than CB" [2]. O'Donovan and Smyth's [8-9] research on trust in recommender systems defines two trust levels, context-specific and system/impersonal trust to help to create and preserve accuracy and robustness within recommender systems. Ziegler and Golbeck's [16] research into trust and interest similarity focused on the link between trust and a person' interest, concluding that the more trust users have between each other, the more their ratings are similar. Tintarev [13] and Tintarev and Masthoff [14] argue for effective explanations that can increase user trust, help users make good decisions and improve user experience.

Although much of the research is based on improving the algorithms, the literature shows movement towards a focus on the user. Tintarev and Masthoff [14] use of two focus groups to determine how participants would like to be recommended or dissuaded from watching a movie indicate a change in the field towards direct contact with users. Accuracy metrics of algorithms is not enough to determine the true impact on user choices.

3. LIBRARYTHING

Book recommender systems (LibraryThing, GoodReads, BookMooch, Amazon, All Consuming, Shelfari, etc.) allow users to catalogue books, and receive and share recommendations within a social community. Since its launch in 2005, LibraryThing has grown to over 920,000 users with the largest group representing librarians, 45.5 million books have been catalogued, and where some book recommender systems offer a single algorithm, LibraryThing has multiple recommender algorithms [1]. According to the founder, Tim Spalding, "We've got five algorithms so far, and a few more I haven't brought live, or which lie underneath the current ones. … LibraryThing's data is particularly suited to it, the books you own being a much better representation of taste than the books you buy on a given retailer" [11]. It is a robust book recommender system with a strong social network that offers a fertile area for user-centered research.

LibraryThing users can add book titles to their accounts and receive book recommendations directly from LibraryThing algorithms (automatic recommendations) or other users of the website (member recommendations). Member recommendations are submitted through a manual process that allows LibraryThing users to submit recommendations for any book by going to the book's recommendation page. The majority of recommendations are automatic and for each book, LibraryThing offers six types of recommendations: 1) LibraryThing Combined Recommendations, 2) Special Sauce Recommendations, 3) Books with similar tags, 4) People with this book also have... (more common), 5) People with this book also have... (more obscure), and 6) Books with similar library subjects and classification. Most of the titles of the recommendation types are self-explanatory in that a user can easily get the general idea of the type of recommendations being offered. For example, the "LibraryThing Combined Recommendations" represents a combination of other types of automatic recommendations. However, the "Special Sauce Recommendations" seems to be the one title that is not self-explanatory and offers no immediate understanding of what users should expect. Spalding says, "Our Special Sauce Recommendation engine is the only one we don't talk about how it works," [11].

4. RESEARCH DESIGN

This study used an online survey ("LibraryThing Recommendation Impact Survey") to explore the impact of LibraryThing recommendations on user choices. No personal or identifying information was collected. There were 10 questions using both open and closed question types. Two of the ten questions focused on capturing demographic data (gender and age range) so that responses could be grouped within a larger context. The other eight questions focused specifically on LibraryThing recommendations and user preferences, influences and actions. Before administering the survey, permission was obtained from Tim Spalding, and an IRB approval from the University.

4.1 Implementation

On October 27th, 2009, the recruitment letter with a link to the survey was posted to "Book Talk," a LibraryThing group recommended by Tim Spalding as a place for major discussions. Spalding pointed out that postings can be tagged for spamming if posted to multiple groups and the goal was to reach the LibraryThing users rather than have the posting removed. However, after a few weeks within the "Book Talk" group, the posting was added to the "Librarians who LibraryThing" group because they were one of the largest groups of LibraryThing users, which helped with getting survey respondents. The posting was repeatedly checked to make sure that it was still on the first page of the active group discussion and if it wasn't, it was adjusted to remain prominent to improve visibility and opportunity for user response. The survey was posted on LibraryThing for five months, from October, 27th, 2009 to March 27th, 2010.

4.2 Participants

Participants were 18 years and older who have previously or were currently using LibraryThing that volunteered to take the survey by clicking the link to the survey from the LibraryThing group. The expectation was that the survey may receive about 100 self-selected respondents and within the five months, there were 62 survey respondents.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper2.pdf

5. RESULTS

The data gathered from the survey used descriptive statistics to generate percentages and iterative pattern coding of qualitative data to identify major themes [4].

5.1 Demographic

Two demographic questions (gender and age range) helped to frame the population responding to the survey. For gender, there were 50 females (81%) and 12 males (19%) who responded to the survey. All age range groups had at least 3 participants. The 25-34 years old range accounted for 42% (26) of participants and the 45-54 years old range accounted for 26% (16) of participants, representing the two largest groups responding to the survey. Overall, there were no age ranges that had zero participants, but the 55-64 age range was the only group with no male participants.

5.2 Member vs. Automatic Recommendations

In their own words, participants described their preferences regarding automatic and member recommendations, and from the data five participant preference categories were developed: automatic, member, both, neither, and no preference. Of the 62 participants that responded to the survey, the majority 48% (30) preferred member recommendations while only 24% (15) preferred automatic recommendations. The other 28% (17) of the participants preferred neither, both or had no preference (Table 1).

Table 1: User Recommendation Preferences

User Preference	*# of Participants*
Automatic	30 (48%)
Member	15 (24%)
Neither	9 (15%)
Both	4 (6.5%)
No Preference	4 (6.5%)

In addition, there was an even split of participants (50%) between those who have submitted member recommendations and those who have not. Participants were also asked to identify their preference for a specific type of LibraryThing automatic recommendation, the top two preferences were "LibraryThing Combined Recommendations" and "People with this book also have…. (more common)" (Table 2).

Table 2: Users' Most Valuable Automatic Recommendations

# of Participants	*Automatic Recommendation Type*
15	LibraryThing Combined Recommendations
14	People with this book also have… (more common)
12	Other
9	Books with similar tags
5	Special Sauce Recommendations
4	Books with similar library subjects and classification
3	People with this book also have... (more obscure)

5.2.1 Discussion

The data suggested that twice as many participants preferred member recommendations over automatic recommendations. Based on reasons provided by participants, a distinction could be made between preferring member or automatic recommendations. Participants that preferred member recommendations seemed to be interested in the social connection between the recommendation and the recommender where they were able to assess the recommender and recommendation as it relates to their own tastes. As one participant described, "Even though automatic recommendations may more 'accurately measure' my tastes and interests based upon the books I have in my library, I feel recommendations from real human beings have the advantage of the recommender's intuitive understanding of what I would find interesting based upon their own impressions of books they know I've read." Alternatively, participants that preferred automatic recommendations seem to be interested in the logical connection of the recommendation and user libraries where the algorithm looks at all items. As one participant stated, "I prefer automatic recommendations because they are based on all users with a particular book, not just on one member who thinks a book is like another." In both cases, the preference for member or automatic recommendations is influenced by the user's trust in particular aspects of the system, which has an impact on the level of trust that the user has of the system and their fellow users. Research into trust models such as a user's trust in another user based on that other user's profile or a user's trust in the system based on the items can begin to offer another dimension for developing recommendations [8-9].

The top preferences for automatic recommendations (Table 2) suggest that LibraryThing users want recommendation types that are additionally filtered (combined recommendations) and socially connected (people also have). The other preferences suggest that there may be overlap with the combined recommendations, lack of knowledge ("What is special sauce? I missed that!"), or an alternative approach to getting recommendations ("People whose library is similar to mine," "Top 1,000 on my recommendations page," "The stars, recommendations in forums").

Since automatic and member recommendations present different ways of getting recommendations within the system, as expected, Table 1 shows that some participants preferred both (6.5%) or had no preference (6.5%). However, the neither category suggested that participants (15%) actively did not prefer automatic or member recommendations, but instead, preferred to get their recommendations from other sources such as message boards ("message boards on the site--it's much more useful for me to read another member's opinion about a book or to see a dialogue about a book on the message boards than to just see a list") or chat ("The recommendations that I DO pay attention to, however, are the ones made personally from people I regularly chat with on LT, and whose tastes I know I share"). The neither category presents an opportunity to understand why some participants are not using the traditional automatic and member recommendations, and how recommender systems can be improved to service this population that seeks alternative methods of getting recommendations that combine multiple sources. These results also suggest looking at the overall goal of the recommender system to identify how best to guide users and filter content appropriately to satisfy user wants and needs [6].

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper2.pdf

5.3 Recommendation Impact

Users were asked if they checked their recommendations, what they did with the information, and how it influenced their choices. Table 3 shows that only 8 (13%) participants never checked their recommendations while 46 (74%) participants checked their recommendations daily, weekly or periodically. Most of the 8 (13%) participants that chose "Other" checked their recommendations on a different schedule than what was presented in the survey question.

Table 3: Frequency of User Checking Recommendations

User Checks	*# of Participants*
Periodically	22 (35%)
Weekly	15 (24%)
Daily	9 (15%)
Other	8 (13%)
Never	8 (13%)

After checking their recommendations, 61% (38) of participants read and followed-up on recommendations (Table 4).

Table 4: Participant Follow-up on Recommendations

Follow up	*# of Participants*
Read and follow-up on recommendations	38 (61%)
Only read recommendations	6 (10%)
Never read or follow-up on recommendations	9 (15%)
Other	9 (15%)

Participants were asked to select specific actions that they took as a result of recommendations and could select multiple responses to indicate the types of influence the recommendations had on their choices. As a result, there were 167 responses, which exceed the number of participants (62), with an average of 2.7 responses per participant. Table 5 shows the selection options and the number of responses per selection.

Table 5: Recommendation Influence

Recommendation Influence	*# of Responses*
Added books to my library.	36
Purchased the recommended book or added to a list for purchase.	35
Browsed user libraries that have the recommended book	31
Reminded you of something else.	29
Submitted a recommendation.	19
Other	17

5.3.1 Discussion

It was important to know whether users were actively engaging the recommender system or taking a passive approach by just reading whatever appears on the homepage. The data show that a majority of the participants checked whether they had new LibraryThing recommendations (Table 3) and followed up on those recommendations by adding books to their libraries, purchasing recommended books or putting recommended books on a list to purchase, and browsed other user libraries with recommended book (Table 4). Table 5 shows 17 "Other" responses, suggesting a need for additional options for users to describe the influences of LibraryThing recommendations, such as no influence, added to wishlist within or outside of LibraryThing, borrowed from local library, and discovery research leading to additional information. Most participants, 46 (74%), found LibraryThing recommendations useful and stated that the recommendations helped them to find books they would not have found otherwise. One participant pointed out the international nature of LibraryThing, "Useful as an introduction to unknown authors and series - particularly American titles - often difficult to source in the UK." Nine (15%) participants found the recommendations "somewhat" useful, and 7 (11%) participants did not find recommendations useful. One participant stated, "I suppose I feel the recommendations function is less useful because it doesn't account for shifting literary interests," highlight an issue for user satisfaction and perceived usefulness.

Perceived usefulness is another area of research that can help to shed light on recommender systems from the user's perspective. Swearingen and Sinha's [12] research comparing online and offline recommendations, focused on perceived usefulness and found that what mattered most was whether users got useful recommendations, the reason for using the recommender system. Overall, LibraryThing participants checked, followed, acted upon and found useful the recommendations they received from LibraryThing and on multiple questions, indicated the impact of recommendations on their choices.

6. LIMITATIONS & FUTURE

One limitation of this study is the self-selected nature of the online survey, which limits the respondents to frequent users of LibraryThing who chose to respond to the survey. This can create a self-selected group of users that do not represent the full range of LibraryThing users. As a consequence, the results are not easily generalized to the larger population and an exploratory survey only scratches the surface of the user perspective. However, this research provides a valuable starting point for future research into user experience with recommender systems, particularly focusing on user preference, user actions and perceived usefulness of recommendations. Based on the themes identified, future research would include creating a more robust method of soliciting data directly from users and in-depth analysis of the "other" categories identified as these categories seem to indicate that users are using the system in unexpected ways, which in turn can help to improve recommender systems.

7. CONCLUSION

The main research goal of this study was to explore the impact of recommendations through recommender systems on user choices and behaviors, particularly what users did as a result of getting a recommendation. Much of the literature on evaluation has focused on the algorithms [5-6], but research into trust [8-9, 16], explanations [13-14], design and usefulness [12]

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper2.pdf

are getting closer to the user of the system. Understanding impact directly from users is an important aspect of developing recommender system research on evaluation and this study has contributed to this effort.

For LibraryThing, the results from this exploratory study indicate possible areas of improvement such as limiting automatic recommendation types because participants preferred only 2-3 out of 6 automatic recommendation types, improving submission of member recommendations because twice as many participants preferred member recommendations over automatic recommendations, and providing alternative recommendations from other areas of LibraryThing because participants indicated a growing need to get recommendations from alternative sources such as tags, message boards, and other areas of LibraryThing.

The research has shown that twice as many participants preferred member recommendations over automatic recommendations, and participants checked, followed-up, acted upon and found recommendations useful. The findings indicate that there's more to uncover within the evaluation of recommender system and that users are an important aspect of understanding whether recommender systems are indeed useful and impactful in people's daily lives.

8. ACKNOWLEDGMENTS

We thank Tim Spalding, LibraryThing Founder, and IMLS fellowship funding for making this research study possible.

9. REFERENCES

[1] *LibraryThing*. Available from: http://www.librarything.com.
[2] Celma, O. and P. Herrera, *A new approach to evaluating novel recommendations*, in *Proceedings of the 2008 ACM conference on Recommender systems*. 2008, ACM: Lausanne, Switzerland. p. 179-186.
[3] Felfernig, A. and R. Burke, *Constraint-based recommender systems: technologies and research issues*, in *Proceedings of the 10th international conference on Electronic commerce*. 2008, ACM: Innsbruck, Austria.
[4] Glaser, B.G. and A.L. Strauss, *The constant comparative method of qualitative analysis*, in *The discovery of grounded theory: Strategies for qualitative research*. 1967, Aldine de Gruyter: Hawthorne, NY. p. 101-115.
[5] Herlocker, J.L., et al., *Evaluating collaborative filtering recommender systems.* ACM Trans. Inf. Syst., 2004. **22**(1): p. 5-53.
[6] Hernández del Olmo, F. and E. Gaudioso, *Evaluation of recommender systems: A new approach.* Expert Systems with Applications, 2008. **35**(3): p. 790-804.
[7] Huang, Z., et al., *A graph-based recommender system for digital library*, in *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*. 2002, ACM: Portland, Oregon, USA. p. 65-73.
[8] O'Donovan, J. and B. Smyth, *Trust in recommender systems*, in *Proceedings of the 10th international conference on Intelligent user interfaces*. 2005, ACM: San Diego, California, USA. p. 167-174.
[9] O'Donovan, J. and B. Smyth, *Is trust robust?: an analysis of trust-based recommendation*, in *Proceedings of the 11th international conference on Intelligent user interfaces*. 2006, ACM: Sydney, Australia. p. 101-108.
[10] Resnick, P. and H.R. Varian, *Recommender systems.* Commun. ACM, 1997. **40**: p. 56-58.
[11] Starr, J., *LibraryThing.com: The Holy Grail of Book Recommendation Engines*, in *Searcher*. 2007. p. 25-32.
[12] Swearingen, K. and R. Sinha. *Beyond Algorithms: An HCI Perspective on Recommender Systems*. in *Proceedings in the SIGIR 2001 Workshop on Recommender Systems*. 2001.
[13] Tintarev, N., *Explanations of recommendations*, in *Proceedings of the 2007 ACM conference on Recommender systems*. 2007, ACM: Minneapolis, MN, USA.
[14] Tintarev, N. and J. Masthoff, *Effective explanations of recommendations: user-centered design*, in *Proceedings of the 2007 ACM conference on Recommender systems*. 2007, ACM: Minneapolis, MN, USA. p. 153-156.
[15] Torres, R., et al., *Enhancing digital libraries with TechLens+*, in *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*. 2004, ACM: Tuscon, AZ, USA. p. 228-236.
[16] Ziegler, C.-N. and J. Golbeck, *Investigating interactions of trust and interest similarity.* Decision Support System, 2007. **43**(2): p. 460-475.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

A User-Centric Evaluation Framework of Recommender Systems

Pearl Pu
Human Computer Interaction Group
Swiss Federal Institute of Technology (EPFL)
CH-1015, Lausanne, Switzerland
Tel: +41-21-6936081
pearl.pu@epfl.ch

Li Chen
Department of Computer Science
Hong Kong Baptist University
224 Waterloo Road, Hong Kong
Tel: +852-34117090
lichen@comp.hkbu.edu.hk

ABSTRACT
User experience research is increasingly attracting researchers' attention in the recommender system community. Existing works in this area have suggested a set of criteria detailing the characteristics that constitute an effective and satisfying recommender system from the user's point of view. To combine these criteria into a more comprehensive framework which can be used to evaluate the perceived qualities of recommender systems, we have developed a model called ***ResQue*** (***Re***commender ***s***ystems' ***Q***uality of ***u***ser ***e***xperience). ResQue consists of 13 constructs and a total of 60 question items, and it aims to assess the perceived qualities of recommenders such as their usability, usefulness, interface and interaction qualities, users' satisfaction of the systems, and the influence of these qualities on users' behavioral intentions, including their intention to purchase the products recommended to them, return to the system in the future, and tell their friend about the system. This model thus identifies the essential qualities of an effective and satisfying recommender system and the essential determinants that motivate users to adopt this technology. The related questionnaire can be further adapted for a custom-made user evaluation or combined with objective performance measures. We also propose a simplified version of the model with 15 questions which can be employed as a usability questionnaire for recommender systems.

Categories and Subject Descriptors
H1.2 [**User/Machine Systems**]: *Human factors;* H5.2 [**User Interfaces**]: *evaluation/methodology, user-centered design.*

General Terms
Measurement, Experimentation, Human Factors.

Keywords
Quality measurement, usability evaluation, recommender systems, quality of user experience, e-Commerce recommender, post-study questionnaire, evaluation of decision support.

1. INTRODUCTION
A recommender system is a web technology that proactively suggests items of interest to users based on their objective behavior or their explicitly stated preferences. It is no longer a fanciful website add-on, but a necessary component. According to the 2007 ChoiceStream survey,[1] 45% of users are more likely to shop at a website that employs recommender technology. Furthermore, a higher percentage (69%) of users in the highest spending category are more likely to desire the support of recommendation technology.

Characterizing and evaluating the quality of user experience and users' subjective attitudes toward the acceptance of recommender technology is an important issue which merits attention from researchers and practitioners in both web technology and human factor fields. This is because recommender technology is becoming widely accepted as an important component that provides both user benefits and enhances the website's revenue. For users, the benefits include more efficiency in finding preferential items, more confidence in making a purchase decision, and a potential chance to discover something new. For the marketer, this technology can significantly enhance user likelihood to buy the items recommended to them, their overall satisfaction and loyalty, increasing users' likelihood to return to the site and recommend the site to their friends. Thus, evaluating user's perception of a recommender system can help developers and marketers understand more precisely if users actually experience and appreciate the intended benefits. This will, in turn, help improve the various aspects of the system and more accurately predict the adoption of a particular recommender.

So far, previous research work on recommender system evaluation has mainly focused on algorithm accuracy [9,1], especially objective prediction accuracy [25,26]. More recently, researchers began examining issues related to users' subjective opinions [30, 13] and developing additional criteria to evaluate recommender systems [18, 33]. In particular, they suggest that user satisfaction does not always correlate with high recommender accuracy. Increasingly, researchers are investigating user experience issues such as identifying determinants that influence users' perception of recommender systems [30], effective preference elicitation methods [19], techniques that motivate users to rate items that they have experienced [2], methods that generate diverse and more satisfying recommendation lists [43], explanation interfaces [31], trust formation with recommenders [6], and design guidelines for enhancing a recommender's interface layout [22]. However, the field lacks a general definition and evaluation framework of what constitutes an effective and satisfying recommender system from the user's perspective.

UCERSTI Workshop of RecSys'10, Sept. 26-30, 2010, Barcelona, Spain.

[1] 2007 ChoiceStream Personalization Survey, ChoiceStream, Inc.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

Our present work aims to review existing usability-oriented evaluation research in the field of recommender systems to identify essential determinants that motivate users to adopt this technology. We then apply well-known usability evaluation models, including TAM [7] and SUMI [15], in order to develop a more balanced framework. The final model, which we call ResQue, consists of 13 constructs and a total of 60 question items categorized into four main dimensions: the perceived system qualities, users' beliefs as a result of these qualities, their subjective attitudes, and their behavioral intentions. The structure and criteria of our framework is derived on the basis of three essential characteristics of recommender systems: 1) being an interaction-driven application and a critical part of online e-commerce services, 2) providing information filtering technology and suggesting recommended items, and 3) providing decision support technology for the users.

The main contribution of this paper is the development of a well-balanced evaluation framework for measuring the perceived qualities of a recommender and predicting users' behavioral intentions as a result of these qualities. Thus, it is a forecasting model that helps us understand users' motivation in adopting a certain recommender. Secondly, the framework aims to help designers and researchers easily perform a usability and user acceptance test during any stage of the design and deployment phase of a recommender. These usability tests can be performed either on a stand-alone basis or as a post-study questionnaire. The model can be further combined with measurements that address other perceived qualities of a recommender, such as security and robustness issues. For those who are interested in a quick usability evaluation, we also propose a simplified version of the model with 15 questions.

2. EVALUATION WORK FROM USERS' POINT OF VIEW

Swearingen and Sinha [38] conducted a user study on eleven recommender systems in order to understand and discover influential factors, other than algorithm accuracy, that affect users' perception. The main results are that transparent system logic, recommendation of familiar items, and sufficient supporting information to recommended items is crucial in influencing users' favorable perception towards the system. They also highlighted that trust and willingness to purchase should be noted. In addition, the users' appreciation of online recommendations is compared with that of recommendations from their friends, defining the notion of relative accuracy.

McNee et al. [20] pointed out that accuracy metrics alone and the commonly employed leave-one-out procedure was very limited in evaluating recommender systems. User satisfaction does not always correlate with high recommender accuracy. Metrics are needed to determine good and useful recommendations, such as the serendipity, salience, and diversity of the recommended items.

Tintarev and Masthoff provided a comprehensive survey of the explanation functionality used in ten academic and eight commercial recommenders [31]. They derived seven main aims of the explanation facility which can help a recommender significantly enhance users' satisfaction: transparency (explains why recommendations were generated), scrutability (the ability for the user to critique the system), trust (increase users' confidence in the system), effectiveness (help users make good decisions), persuasiveness (convince users to try or buy items recommended to them), efficiency (help users make decisions faster) and satisfaction (increase the ease of use and enjoyment). These aims are very similar to the set of criteria that we have developed in ResQue, except the fact that we focus more on the system as a whole rather than just the explanation component.

Ozok et al. [22] explored recommender systems' usability and user preferences from both the structural (how recommender systems should look) and content (what information recommender systems should contain) perspectives. A two-layer interface usability evaluation model including both micro- and macro-level interface evaluations was proposed, followed by a Survey on Usability of E-Commerce Recommender Systems (SUERS). The survey was administered on 131 college-aged online shoppers to measure and rank the importance of structural and content aspects of recommender systems from the shoppers' perspectives. The main result was a set of 14 design guidelines. The micro-level of the guidelines provided suggestions specific to the recommended product such as what attributes (name, price, image, description, rating, etc.) to include in the interface. The macro-level of the guidelines provided suggestions concerning when, where and how the recommended products should be displayed. The development process of the model was limited, as authors did not go through an iterative process of the evaluation and refinement of the model. Instead, it was purely based on a literature survey of quite limited past work of subjective evaluations of recommender system. Most importantly, it failed to explain how usability issues influence users' behavioral intentions such as their intention to buy the items recommended to them, whether they will continue using the system and recommend the system to their friends.

Jones and Pu [13] presented the first significant user study that aimed to understand users' *initial* adoption of the recommender technology and their subjective perceptions of the system. Study results show that a simple interface design, a small amount of initial effort required by the system to get to know the users, the perceived qualities such as the subjective accuracy, novelty and enjoyability of the recommended items are the key design factors that significantly enhance the website's ability to attract users.

3. MODEL DEVELOPMENT

A measurement model consists of a set of constructs, the participating questions for each construct, the scale's dimensions, and a procedure for conducting the questionnaire. Psychometric questionnaires such as the one proposed in this paper require the validation of the questions used, data gathering, and statistical analysis before they can be used with confidence. The current model and its constructs were based on our past work in investigating various interface and interaction issues between users and recommenders. In over 10 user studies, we have carefully and progressively developed and employed user satisfaction questionnaires to evaluate recommenders' perceived qualities such as ease of use, perceived usefulness and users' satisfaction and behavioral intentions [4,5,6,12,13,14,23,24]. This past research has given us a unique opportunity to synthesize and organize the accumulation of existing questionnaires and develop a well-balanced framework.

In the model development process, we also compare our constructs with those used in TAM and SUMI, two well-known and widely adopted measurement frameworks.

TAM (Technology Acceptance Model) seeks to understand a set of perceived qualities of a system and users' intention to adopt the system as a result of these qualities, thus explaining not only the desirable outcome of a system but also users' motivation. The

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

original TAM listed three constructs: perceived ease of use of a system, its perceived usefulness and users' intention to use the system. However, TAM was also criticized for its over-simplicity and generality. Venkatesh et al. [32] formulated an updated version of TAM, called the Unified Theory of Acceptance and Use of Technology. In this more recent theory, four key constructs (performance expectancy, effort expectancy, social influence, and facilitating conditions) were presented as direct determinants of usage intentions and behaviors.

SUMI (Software Usability Measurement Inventory) is a psychometric evaluation model developed by Kirakowski and Corbett [15] to measure the quality of software from the end-user's point of view. The model consists of 5 constructs (efficiency, affect, helpfulness, control, learnability) and 50 questions. It is widely used to help designers and developers assess the quality of use of a software product or prototype and can assist with the detection of usability flaws and the comparison between software products.

By adapting our past work to the TAM and SUMI models, we have identified 4 essential constructs of ResQue for a successful recommender system to fulfill from the users' point of view: 1) user perceived qualities of the system, 2) user beliefs as a result of these qualities in terms of ease of use, usefulness and control, 3) their subjective attitudes, and 4) their behavioral intentions. Figure 1 depicts the detailed schema of the constructs of ResQue and some of the scales for each construct.

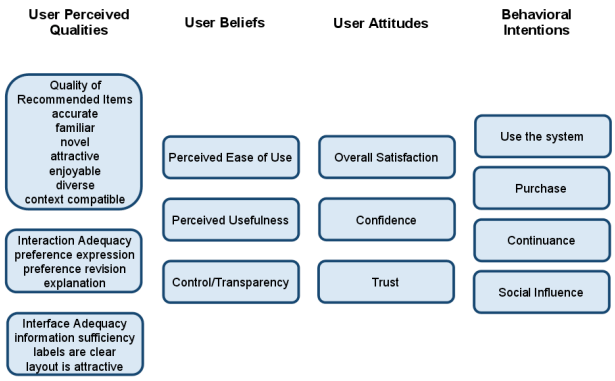

Figure 1: Constructs of an Evaluation Framework on the Perceived Qualities of Recommenders (ResQue).

When administering the questionnaires, we assume that a recommender system being evaluated is part of an online system. To make the evaluation more focused on the recommender component, we often give subjects a specific task: "*find an ideal product to buy/experience from an online site*" where the recommender in question is a constituent component.

In the following sections, the meaning of each scale as well as its subscales is defined and explained, and the sample questions that can be used in a questionnaire are suggested in the appendix at the end of the paper. It is a common practice in questionnaire development to vary the tone of items to control potential response biases. Typically some of the items elicit agreement and others elicit disagreement. For some of the items, therefore, we also suggest reverse scale questions. A 5-point Likert scale from "strongly disagree" (1) to "strongly agree" (5) is recommended to characterize users' responses.

3.1 Perceived System Qualities

This construct refers to the functional and informational aspect of a recommender and how the perceived qualities of these aspects influence users' beliefs on the ease of use, usefulness and control/transparency of a system. A recommender system is not simply part of a website, but more importantly a decision support tool. We focus on three essential dimensions: the quality of the recommended items, the interaction adequacy and the interface adequacy as the recommender helps users reach a purchase decision.

3.1.1 Quality of Recommended Items

The items proposed by a recommender can be considered one of the main features of the system. Qualities refer to the information quality and genuine usefulness of the suggested items. Presented as a collection of articles, the recommended items are often labeled and presented in a certain area of the recommender page. Some systems also propose grouping them into meaningful subareas to increase users' comprehension of the list and enable them to more effectively reach decisions [4]. In our earlier work, we have found strong correlations of the following qualities of the recommended items to users' intention to use the system.

Perceived accuracy is the degree to which users feel the recommendations match their interests and preferences. It is an overall assessment of how well the recommender has understood the users' preferences and tastes. This subjective measure is significantly easier to obtain than the measure of objective accuracy that we used in our earlier work [23]. Our studies show that they are strongly correlated [6]. In other words, if users respond well to this question, it is likely that the underlying algorithm is accurate in predicting users' interest. In addition, it is useful to use **relative accuracy** to compare the difference between recommendations a user may get from a system versus those from friends [28]. It can serve as a useful complement to perceived accuracy because it implicitly sets up friends' recommendation quality as a baseline.

Familiarity describes whether or not users have previous knowledge of, or experience with, the items recommended to them. Swearingen and Sinha [30] indicated that users like and prefer to get recommendations of previously experienced items because their presence reinforces trust in the recommender system. However, users can be frustrated by too much familiarity. Therefore, it is important to know whether or not a recommender website has achieved the proper balance of familiarity and novelty from the users' perspective.

Novelty (or discovery) is the extent to which users receive new and interesting recommendations. The core concept of novelty is related to the recommender's ability to educate users and help them discover new items [24]. In [20], a similar concept, called "serendipity", was suggested. Herlocker [11] argued that novelty is different from serendipity, because novelty only covers the concept of "new" while serendipity means not only "new" but also "surprising". However, in conducting the actual user evaluation procedure, the meticulous distinction between these two words will cause confusion for users. Therefore, we suggest novelty and discovery as two similar questions. More user trials will be needed to further delineate the serendipity question.

The **Attractiveness** of the recommended items refers to whether or not recommended items are capable of stimulating users' imagination and evoking a positive emotion of interest or desire.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

Attractiveness is different from accuracy and novelty. An item can be accurate and novel, but not necessarily attractive; a novel item is different from anything a user has ever experienced, whereas an attractive item stimulates the user in a positive manner. This concept is similar to the salience factor in [20].

While judging novelty requires a user to think more about the distinguishing factors of an item, the aspect of attractiveness brings to mind the outstanding quality of an item and has a more emotional tone to it.

The **enjoyability** of recommended items refers to whether users have enjoyed experiencing the items suggested to them. It was found to have a significant correlation to users' intention to use and return to the system [13]. This is the only scale that assesses a user's actual experience of a recommender. In many online study scenarios, it is not possible to immediately measure enjoyability unless users are told to answer a questionnaire after a few weeks when they have actually received and experienced the item. In testing music or film recommenders, it is possible to allow users to answer this question if they are given the opportunity to listen to a song excerpt or watch a movie trailer.

Diversity measures the diversity level of items in the recommendation list. As the recommendation list is the first piece of information users will encounter before they examine the details of an individual recommendation, users' impression of this list is important for their perception of the whole system. At this stage, it has been found that a low diversity level might disappoint users and could cause them to leave this recommender [13]. McGinty and Smyth [17] proposed integrating diversity with similarity in order to adaptively select the appropriate strategy (either similar or diverse ones) given each individual user's past behavior and current needs. Literature also suggests that a recommendation list as a complete entity should be judged for its diversity rather than treating each recommendation as an isolated item [33].

Context compatibility evaluates whether or not the recommendations consider general or personal context requirements. For example, for a movie recommender, the necessary context information may include a user's current mood, different occasions for watching the movie, whether or not other people will be present, and whether the recommendation is timely. A good recommender system should be able to formulate recommendations considering different kinds of contextual factors that will likely take effect.

3.1.2 Interaction Adequacy

Besides issues related to the quality of recommended items, the system's ability to present recommendations, to allow for user feedback and to explain the reasons why recommendations facilitate purchasing decisions also weighs highly on users' overall perception of a recommender. Thus, three main interaction mechanisms are usually suggested in various recommenders: initial preference elicitation, preference revision, and the system's ability to explain its results. Behavioral based recommenders do not require users to explicitly indicate their preferences, but collect such information via users' browsing and purchasing history. For rating and preference based recommenders, this process requires a user to rate a set of items or state their preferences on desired items in a graphical user interface [23]. Some conversational recommenders provide explicit mechanisms for users to provide feedback in the form of critiques [6]. The simplest critiques indicate whether the recommended item is good or bad, while the more sophisticated ones show users a set of alternative items that take into account users' desire for these items and the potential superior values they offer, such as better price, more popularity, etc [6].

The final interaction quality being measured is the system's ability to explain the recommended results. Herlocker et al. [10], Sinha and Swearingen [30] and Tintarev and Masthoff [31] demonstrated that a good explanation interface could help inspire users' trust and satisfaction by giving them information to personally justify recommendations, increasing user involvement and educating users on the internal logic of the system [10, 31]. In addition, Tintarev and Masthoff [31] defined in detail possible aims of explanation facilities: transparency, scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction. Pu and Chen extensively investigated design guidelines for developing explanation-based recommender interfaces [4]. They found that organization interfaces are particularly effective in promoting users' satisfaction of the system, convincing them to buy items recommended to them, and bringing them back to the store in the future.

3.1.3 Interface Adequacy

Interface design issues related to recommenders have also been extensively investigated in [10, 20, 31,22]. Most of the existing work is concerned with how to optimize the recommender page layout to achieve the maximum visibility of the recommendation, i.e. whether to use image, text, or a combination of the two. A detailed set of design guidelines were investigated and proposed [22]. In our current model, we mainly emphasize users' subjective evaluations of a recommender interface in terms of its information sufficiency, the interface label and layout adequacy and clarity.

3.2 Beliefs

3.2.1 Perceived Ease of Use

Perceived ease of use, also known as efficiency in SUMI and perceived cognitive effort in our existing work [6,14], measures users' ability to quickly and correctly accomplish tasks with ease and without frustration. We also use it to refer to decision efficiency, i.e. the extent to which a recommender system facilitates users to find their preferential items quickly. Although task completion and learning time can be measured objectively, it can be difficult to distinguish the actual task completion time from the measured task time for various reasons. Users can be exploring the website and discovering information unrelated to the assigned task. This is especially true if a system is entertaining and educational, and its interface and content is very appealing. It is also possible that the user perceives that he/she has consumed less time while the measured task completion time is in fact high. Therefore, evaluating perceived ease of use may be more appropriate than using the objective task completion time to measure a system's ease of use.

Besides the overall perceived ease of use, **perceived initial effort** should also be taken into account, given the new user problem. Perceived initial effort is the perceived effort users contribute to the system before they get the first set of recommendations. The initial effort could be spent on rating items [19], specifying preferences, or answering personality quizzes [12]. Theoretically speaking, recommender systems should try to minimize the effort users expend for a good recommendation [30].

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

Easy to learn, known as "learnability" in SUMI, initially appears to be an inadequate dimension since most recommenders require a minimal amount of learning by design. However, since some users may not initially notice the recommended items or know exactly what they were intended for, especially without clear labels or explicit explanations on the interface, the learning aspect should be included to measure the level of ease for users to discover the recommended items. In addition, some recommenders, such as critiquing-based recommenders, do allow users to provide feedback to increase the personalization of the recommender. In this case, the learning construct measures how easy it is for users to alter their personal profile information in order to receive different recommendations.

3.2.2 Perceived Usefulness

Perceived usefulness of a recommender (called perceived competence in our previous work) is the extent to which a user finds that using a recommender system would improve his/her performance, compared with their previous experiences without the help of a recommender [4]. This element requests users' opinion as to whether or not this system is useful to them. Since recommenders used in e-commerce environments mainly assist users in finding relevant information to support their purchase decision, we further qualify the usefulness in two aspects: decision support and decision quality.

Recommender technology provides decision support to users in the process of selecting preferential items, for example making a purchase in an e-commerce environment. The objective of decision technologies in general is to overcome the limit of users' bounded rationality and to help them make more satisfying decisions with a minimal amount of effort [29]. Recommender systems specifically help users manage an overwhelming flood of information and make high-quality decisions under limited time and knowledge constraints. **Decision support** thus measures the extent to which users feel assisted by the recommended system.

In addition to the efficiency of decision making, the quality of the decision (**decision quality)** also matters. The quality of a system-facilitated decision can be assessed by confidence criterion, which is the level of a user's certainty in believing that he/she has made a correct choice with the assistance of a recommender.

3.2.3 Control and Transparency

User control measures whether users felt in control in their interaction with the recommender. The concept of user control includes the system's ability to allow users to revise their preferences, to customize received recommendations, and to request a new set of recommendations. This aspect weighs heavily in the overall user experience of the system. If the system does not provide a mechanism for a user to reject recommendations that he/she dislikes, a user will be unable to stop the system from continuously recommending items which might cause him/her to be disappointed with the system.

Transparency determines whether or not a system allows users to understand its inner logic, i.e. why a particular item is recommended to them. A recommender system can convey its inner logic to the user via an explanation interface [4,10,30,31]. To date, many researchers have emphasized that transparency has a certain impact on other critical aspects of users' perception. Swearingen and Sinha [30] showed that the more transparent a recommended product is, the more likely users would be to purchase it. In addition, Simonson [27] suggested that perceived accuracy of a recommendation is dependent on whether or not the user sees a correspondence between the preferences expressed in the measurement process and the recommendation presented by the system.

3.3 Attitudes

Attitude is a user's overall feeling towards a recommender, which is most likely derived from his/her experience as she interacts with a recommender. An attitude is generally believed to be more long-lasting than a belief. Users' attitudes towards a recommender are highly influential on their subsequent behavioral intentions. Many researchers attribute positive attitudes, including users' satisfaction and trust of a recommender, as important factors.

Evaluating **overall satisfaction** determines what users think and feel while using a recommender system. It gives users an opportunity to express their preferences and opinions about a system in a direct way. **Confidence inspiring** refers to the recommender's ability to inspire confidence in users, or its ability to convince users of the information or products recommended to them. **Trust** indicates whether or not users find the whole system trustworthy. Studies show that consumer trust is positively associated with their intentions to transact, purchase a product, and return to the website [8]. The trust level is determined by the reputation of online systems [8], as well as the recommender system's ability to formulate good recommendations and provide useful explanation interfaces [4,10,19]. However, as trust is a long-term relationship between a user and an online system, it is sometimes difficult to measure trust purely after a short-period interaction with a system. Thus, we recommend observing the trust formation over time, as users are incrementally exposed to the same recommender.

3.4 Behavioral Intentions

Behavioral intentions towards a system is related to whether or not the system is able to influence users' decision to use the system and purchase some of the recommended results.

One of the fundamental goals for an e-commerce website is to maximize user loyalty and the lifetime value to stimulate users' future visits and purchases. User loyalty evaluates the system's ability to convince users to reuse the system, or persuade them to introduce the system to their friends in order to increase the number of clients. Accordingly, this dimension consists of the following criteria: user agreement to use the system, user acceptance of the recommended items (resulting in a purchase), user retention and **intention to introduce this system to her/his friends**. By using a questionnaire, the user's **intention to return** can be measured as a satisfactory approximation of actual user retention, because the Theory of Planned Behavior [32] states that behavioral intention can be a strong predictor of actual behavior. Although the website's integrity, reputation and price quality will also likely impact user loyalty, the most important factor for a recommender system is to help users effectively find a satisfying product, i.e. the quality of its recommendations [7].

4. SIMPLIFIED MODEL

In the previous sections, we described the ***development process*** of a subjective evaluation framework to measure users' perceived qualities of a recommender as well as users' behavioral intentions such as their intention to buy or use the items suggested to them, continue to use the system, and tell their friends about the recommender. We described both the constructs and corresponding sample questions (see Appendix A for a summary).

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

Our overall motivation for this research was to understand the crucial factors that influence the user adoption of recommenders. Another motivation is to come up with a subjective evaluation questionnaire that other researchers and practitioners can employ. However, it is unlikely that a 60-item questionnaire can be administered for a quick and easy evaluation. This has motivated us in proposing a simplified model based on our past research. Between 2005 and 2010, we have administered 11 subjective questionnaires on a total of 807 subjects [4,5,6,12,13,14,23,24]. Initial questionnaires covered some of the four categories identified in the ResQue. As we conducted more experiments, we became more convinced of the four categories and used all of them in recent studies. On average, between 12 and 15 questions were used. Based this previous work, we have synthesized and organized a total of 15 questions as a simplified model for the purpose of performing a quick and easy usability and adoption evaluation of a recommender (see questions with * sign).

5. CONCLUSION AND FUTURE WORK

User evaluation of recommender systems is a crucial subject of study that requires a deep understanding, development and testing of the right dimensions (or constructs) and the standardization of the questions used. The framework described in this paper presents the first attempt to develop a complete and balanced evaluation framework that measures users' subjective attitudes based on their experience towards a recommender.

ResQue consists of a set of 13 constructs and 60 questions for a high-quality recommender system from the user point of view and can be used as a standard guideline for a user evaluation. It can also be adapted to a custom-made user evaluation by tailoring it in an individual research context. Researchers and practitioners can use these questionnaires with ease to measure users' general satisfaction with recommenders, their readiness to adopt the technology, and their intention to purchase recommended items and return to the site in the future.

After ResQue was finalized, we asked several expert researchers in the community of recommender systems to review the model. Their feedback and comments were then incorporated into the final version of the model. This method, known as the Delphi method, is one of the first validation attempts on the model. Since the work was submitted, we have started conducting a survey to further validate the model's reliability, validity and sensitivity using factor analysis, structural equation modeling (SEM), and other techniques described in [21]. Initial results based on 150 participants indicate how the model can be interpreted and show factors that correspond to the original model. At the same time, analysis also gives some indications on how to refine the model. More users are expected to participate in the survey and the final outcome will be soon reported.

APPENDIX

A. Constructs and Questions of ResQue

The following contains the questionnaire statements that can be used in a survey. They are developed based on the ResQue model described in this paper. Users should be asked to indicate their answers to each of the questions using the 1-5 Likert scales, where 1 indicates "strongly disagree" and 5 is "strongly agree."

A1. Quality of Recommended Items

A.1.1 Accuracy

- The items recommended to me matched my interests.*
- The recommender gave me good suggestions.
- I am not interested in the items recommended to me (reverse scale).

A.1.2 Relative Accuracy

- The recommendation I received better fits my interests than what I may receive from a friend.
- A recommendation from my friends better suits my interests than the recommendation from this system (reverse scale).

A.1.3 Familiarity

- Some of the recommended items are familiar to me.
- I am not familiar with the items that were recommended to me (reverse scale).

A.1.4 Attractiveness

- The items recommended to me are attractive.

A.1.5 Enjoyability

- I enjoyed the items recommended to me.

A.1.6 Novelty

- The items recommended to me are novel and interesting.*
- The recommender system is educational.
- The recommender system helps me discover new products.
- I could not find new items through the recommender (reverse scale).

A.1.6 Diversity

- The items recommended to me are diverse.*
- The items recommended to me are similar to each other (reverse scale).*

A.1.7 Context Compatibility

- I was only provided with general recommendations.
- The items recommended to me took my personal context requirements into consideration.
- The recommendations are timely**.**

A2. Interaction Adequacy

- The recommender provides an adequate way for me to express my preferences.
- The recommender provides an adequate way for me to revise my preferences.
- The recommender explains why the products are recommended to me.*

A3. Interface Adequacy

- The recommender's interface provides sufficient information.
- The information provided for the recommended items is sufficient for me.
- The labels of the recommender interface are clear and adequate.
- The layout of the recommender interface is attractive and adequate.*

A4. Perceived Ease of Use

A.4.1 Ease of Initial Learning

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

- I became familiar with the recommender system very quickly.
- I easily found the recommended items.
- Looking for a recommended item required too much effort (reverse scale).

A.4.2 Ease of Preference Elicitation

- I found it easy to tell the system about my preferences.
- It is easy to learn to tell the system what I like.
- It required too much effort to tell the system what I like (reversed scale).

A.4.3 Ease of Preference Revision

- I found it easy to make the system recommend different things to me.
- It is easy to train the system to update my preferences.
- I found it easy to alter the outcome of the recommended items due to my preference changes.
- It is easy for me to inform the system if I dislike/like the recommended item.
- It is easy for me to get a new set of recommendations.

A.4.4 Ease of Decision Making

- Using the recommender to find what I like is easy.
- I was able to take advantage of the recommender very quickly.
- I quickly became productive with the recommender.
- Finding an item to buy with the help of the recommender is easy.*
- Finding an item to buy, even with the help of the recommender, consumes too much time.

A5. Perceived Usefulness

- The recommended items effectively helped me find the ideal product.*
- The recommended items influence my selection of products.
- I feel supported to find what I like with the help of the recommender.*
- I feel supported in selecting the items to buy with the help of the recommender.

A6. Control/Transparency

- I feel in control of telling the recommender what I want.
- I don't feel in control of telling the system what I want.
- I don't feel in control of specifying and changing my preferences (reverse scale).
- I understood why the items were recommended to me.
- The system helps me understand why the items were recommended to me.
- The system seems to control my decision process rather than me (reverse scale).

A7. Attitudes

- Overall, I am satisfied with the recommender.*
- I am convinced of the products recommended to me.*
- I am confident I will like the items recommended to me. *
- The recommender made me more confident about my selection/decision.
- The recommended items made me confused about my choice (reverse scale).
- The recommender can be trusted.

A8. Behavioral Intentions

A.8.1 Intention to Use the System

- If a recommender such as this exists, I will use it to find products to buy.

A.8.2 Continuance and Frequency

- I will use this recommender again.*
- I will use this type of recommender frequently.
- I prefer to use this type of recommender in the future.

A.8.3 Recommendation to Friends

- I will tell my friends about this recommender.*

A.8.4 Purchase Intention

- I would buy the items recommended, given the opportunity.*

6. REFERENCES

[1] Adomavicius, G. and Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734-749.

[2] Beenen, G., Ling, K., Wang, X., Chang, K., Frankowski, D., Resnick, P., et al. 2004. Using social psychology to motivate contributions to online communities. In CSCW '04: Proceedings of the ACM Conference On Computer Supported Cooperative Work. New York: ACM Press.

[3] Castagnos, S., Jones, N., and Pu, P. 2009. Recommenders' Influence on Buyers' Decision Process. In proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009), 361-364.

[4] Chen, L. and Pu, P. 2006. Trust Building with Explanation Interfaces. In Proceedings of International Conference on Intelligent User Interface (IUI'06), 93-100.

[5] Chen, L. and Pu, P. 2008. A Cross-Cultural User Evaluation of Product Recommender Interfaces. RecSys 2008, 75-82.

[6] Chen, L. and Pu, P. 2009. Interaction Design Guidelines on Critiquing-based Recommender Systems. User Modeling and User-Adapted Interaction Journal (UMUAI), Springer Netherlands, Volume 19, Issue3, 167-206.

[7] Davis, F.D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quart.* 13 319-339.

[8] Grabner-Kräuter, S. and Kaluscha, E.A. 2003. Empirical research in on-line trust: a review and critical assessment Int. J. Hum.-Comput. Stud. (IJMMS) 58(6), 783-812.

[9] Herlocker, J.L., Konstan, J.A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proc. of ACM SIGIR 1999*, ACM Press (1999), 230-237.

[10] Herlocker, J.L., Konstan, J.A., and Riedl, J. 2000. Explaining collaborative filtering recommendations. CSCW 2000, 241-250.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper3.pdf

[11] Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J. 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5-53.

[12] Hu, R. and Pu, P. Potential Acceptance Issues of Personality-based Recommender Systems. In Proceedings of ACM Conference on Recommender Systems (RecSys'09), New York City, NY, USA, October 22-25, 2009.

[13] Jones, N., and Pu, P. 2007. User Technology Adoption Issues in Recommender Systems. In Proceedings of Networking and Electronic Commerce Research Conference (NAEC2007), 379-394.

[14] Jones, N., Pu, P., and Chen, L. 2009. How Users Perceive and Appraise Personalized Recommendations. Proceedings of User Modeling, Adaptation, and Personalization conference (UMAP09), 461-466.

[15] Kirakowski, J. 1993. SUMI: the Software Usability Measurement Inventory. *British Journal of Educational Technology*, 24 (3) 210-214.

[16] Lewis, J.R. 1993. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use.

[17] McGinty, L. and Smyth, B. On the role of diversity in conversational recommender systems. In *Proceedings of the Fifth International Conference on Case-Based Reasoning (ICCBR'03)*, 2003, 276-290.

[18] McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A. and Riedl, J. On the Recommending of Citations for Research Papers. In *Proc. of ACM CSCW 2002*, ACM Press (2002), 116-125.

[19] McNee, S.M., Lam, S.K., Konstan, J.A., Riedal, J. 2003. Interfaces for eliciting new user preferences in recommender systems. User Modeling 2003, 178-187.

[20] McNee, S.M., Riedl, J., and Konstan, J.A. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. CHI Extended Abstracts 2006,1097-1101.

[21] Nunnally, J. C. 1978. Psychometric Theory.

[22] Ozok, A.A, Fan, Q., Norcio, A.F. 2010. Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: results from a college student population. Behaviour & Information Technology, Volume 29, Issue 1, 57 - 83.

[23] Pu, P., Chen, L., and Kumar, P. 2008. Evaluating Product Search and Recommender Systems for E-Commerce Environments. Electronic Commerce Research Journal, 8(1-2), June,1-27.

[24] Pu, P., Zhou, M., and Castagnos, S. 2009. Critiquing Recommenders for Public Taste Products. In proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009), 249-252.

[25] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000. Analysis of recommendation algorithms for e-commerce. ACM Conference on Electronic Commerce, 158-167.

[26] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. WWW, 285-295.

[27] Simonson, I. 2005. Determinants of customers' responses to customized offers: Conceptual framework and research propositions. Journal of Marketing, 69 (January 2005), 32–45.

[28] Sinha, R. and Swearingen, K. 2001. Comparing Recommendations made by Online Systems and Friends. Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries, 2001.

[29] Stohr, E.A. and Viswanathan, S. 1999. Recommendation systems: Decision support for the information economy. Emerging Information Technologies, K. E. Kendall, Ed. Thousand Oaks, CA: SAGE, 1999, 21-44.

[30] Swearingen, K. and Sinha, R. 2002. Interaction design for recommender systems. In Interactive Systems (DIS2002).

[31] Tintarev, N. and Masthoff, J. 2007. Survey of explanations in recommender systems. ICDE Workshops 2007, 801-810.

[32] Venkatesh,V., Morris, M.G., Davis, G.B. and Davis, F.D. 2003. User acceptance of information technology: Toward a unified view. MIS Quarterly, 2003, 27, 3, 425-478.

[33] Ziegler, C.N., McNee, S.M., Konstan, J.A., and Lausen, G., Improving Recommendation Lists through Topic Diversification. In *Proc. of WWW 2005*, ACM Press (2005), 22-32.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper4.pdf

User-perceived recommendation quality - factoring in the user interface

Mouzhi Ge
TU Dortmund
44221, Dortmund, Germany
mouzhi.ge@tu-dortmund.de

Carla Delgado-Battenfeld
TU Dortmund
44221, Dortmund Germany
carla.delgado@tu-dortmund.de

Dietmar Jannach
TU Dortmund
44221, Dortmund Germany
dietmar.jannach@tu-dortmund.de

ABSTRACT
Most works in the domain of recommender systems focus on providing accurate recommendations. However many recent works have raised the issue that beyond accuracy other aspects such as diversity and novelty also impact the quality of recommendations and the user/customer behavior. This initiative has opened up a new perspective regarding evaluating and improving recommendation techniques, but some challenges are still to be faced. For example, traditional evaluations of recommenders do not take into account the system's interface. While accuracy is a metric somehow uncoupled to the recommenders' interface, other metrics such as diversity and novelty are directly related to it: a user might better perceive a higher degree of diversity and novelty if this is emphasized by its interface. In this paper we discuss the relations between evaluation metrics, the recommender interface and the user-perceived recommendation quality. We present a general guideline to evaluate recommenders from perspectives other than accuracy and propose a general experiment design to investigate the effects of quality factors on recommendations taking into account the system's interface. We also show how the proposed experiment model could be used to experiment with the factors "diversity" and "novelty" and specifically show how these factors can be meaningfully introduced in an experiment. We believe that our current work can be used in future research as a basis example on how to exam the effects of evaluation metrics and the user interface in recommender systems.

Categories and Subject Descriptors
C.4 [**Performance of Systems**]: Measurement techniques.

General Terms
Measurement, Performance, Reliability.

Keywords
Recommender system, experiment design, evaluation metric, diversity, novelty.

1. INTRODUCTION
The main goal of recommender systems is to provide personalized recommendations in order to improve users' satisfaction and assist the users in making decisions. Different recommender systems were developed and used in several domains over the last decades [1] and a variety of recommendation techniques were proposed. Accordingly, various metrics have been proposed to estimate the effectiveness and value of the recommender systems.

Several among the successful recommendation techniques are based on a prediction of the degree to which a user might like an item. Because of this, the traditional evaluation approaches for recommenders are focused on the accuracy of the generated predictions, based for example on the Mean Absolute Error. Such approaches focus on the algorithm used to generate the recommendations, but do not look at the system as a whole. Usually these measurements are done in offline experiments [12] that do not take into account the user interaction with the system. Thus, such evaluations are typically independent of the system's interface and uncoupled from the user experience.

Although it is clear that the accuracy of the recommendations can affect the perceived quality of the system and the customer/user behavior, recent works argue that there are other important aspects we need to take into account [8, 14]. Several aspects of the perceived value of a recommender depend on the user interface and cannot be captured in an offline-experimental setting, in which e.g. only the ratings are available. According to Francisco Martin, who was RecSys09 keynote, up to 50% of the value of recommenders comes from a well-designed interface. Although this hypothesis is not supported by empirical evidence yet, we indeed believe that the interface of a recommender has a strong effect on its perceived value, and also that changes on the interface will affect the user's perception of the recommendations.

A classical example of the impact of the interface in the user perception of the recommendations is the case of serendipitous items in recommendation lists. When implemented inappropriately, unexpected items in the recommendation list may leave the user with the impression that the system does not understand his real needs, and therefore he may stop following the recommendations or even stop using the system. These risks can be reduced by the use of more (visual) explanations/clues regarding the reasons as to why an item was recommended. This has been done by Amazon.com (*http://www.amazon.com*), where one can see different lists of items classified by headers like "Users that bought this also bought that", "your recommendations" and "special offers" [16]. In this manner, the risk of misinterpreting the principle of the recommendation is reduced.

Several authors have already discussed quality factors beyond accuracy that may influence recommendations, e.g. [9, 13] and also how to use these factors to evaluate recommendations [8, 14, 15, 17]. In particular, [7] touched the matter of the advantages of online over offline evaluation strategies. We consider this a very important step towards exploiting the possibilities that different quality factors can bring to recommenders, but at the same time we believe there is a second step to be made: incorporating the interface and user interaction in the evaluations. Indeed, reports on experiments where quality factors were analyzed together with the recommenders interface already appeared on the literature [4, 20]. In this paper we approach this topic directly and discuss a

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper4.pdf

general evaluation approach that incorporates the system interface. Our main point is that there is a strong influence of the interface on the user perception of the quality of the recommendations received, and experiments that neglect this influence may lead to biased conclusions.

The paper is organized as follows. In Section 2 we describe our general model for representing the relationship among recommendation quality factors, user interface and customer behavior. In Section 3 we give an example of how the model can be instantiated into a specific experiment design. Section 4 focuses on how to incorporate the quality factors "novelty" and "diversity" in an experiment. Section 5 presents our conclusions and plans for future work.

2. MODEL CONSTRUCTION

As mentioned above, we argue that the user perception of different recommendation quality factors may be significantly affected by the system interface. Generally, different user interfaces can be used to present the recommendations. Therefore the user interface can be considered a moderator variable that affects the direction and/or strength of the relation between the recommendation quality factors and the customer behavior.

We propose a general model to examine the relationship among recommendation quality factors, user interface and customer behavior. The model is described as follows.

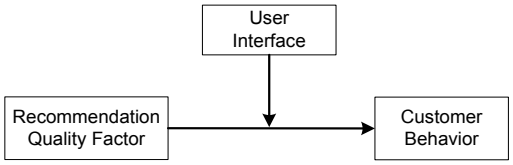

Figure 1: General model of the relationship among recommendation quality factors, user interface and customer behavior.

In our model, "recommendation quality factor" is a general term that represents the several possible factors that indicate different quality aspects of the recommendations. A few factors have been proposed in previous research such as for example diversity, novelty, serendipity and coverage [8, 14]. Also, "user interface" is considered in the context of recommender systems as a display format that allows the customers to interactively explore the recommendations. For example, a recommendation can be visually represented using plain text or a picture (as indicated by [10]). By "customer behavior" we mean the customers' actions or responses that may be affected by recommenders such as customer purchase behavior [2], customer decision making [18], customer interests [19], or satisfaction [20].

The more quality factors we include, the more different interfaces might be used to express the recommendations with different effects on the user perception (it is always the case that different interfaces can be used, but if no quality factors are added there might not be any effects on the user perception). The goal of our model is to analyze the interactions between user interface, quality factors and customer behavior.

When instantiating the model, we are still facing the following questions: How to measure the recommendation quality factors? Which interface can be used to express the recommendations? How to measure customer behavior? In the next section, we develop a first research design of how to implement our model.

3. MODEL INSTANCE AND EXPERIMENT DESIGN

It has been found that experimental research is an effective approach to address cause and effect relationships [3, 11]. In order to show how our general model can be instantiated within a concrete experiment, we selected two well cited evaluation metrics as recommendation quality factors: *diversity* and *novelty,* and use two common interface styles to visualize the recommendations: *single list* and *multiple lists*. The customer behavior is analyzed in terms of *purchase rate* and *customer satisfaction*, since these are the typical indicators for recommender's performance.

Thus, two independent variables are determined, each of which has two possible values: diversity (with or without), novelty (with or without). The values for the variable user interface are also two: single list or multiple lists. The customer behavior is determined by two dependent variables: customer purchase and customer satisfaction. On one hand, the customer purchase is mainly the vendor's perspective and aims at directing customers to adopt or buy the recommended product regardless of their satisfaction. It can be directly measured by the sales increase generated from the recommender system. On the other hand, customer satisfaction stands for the customer perspective and how the recommended products or the recommendation session as a whole fulfilled his expectations. It is usually measured by a survey using Likert scales.

Figure 2 gives an overview of the instantiated model, once the independent and dependent variables are determined.

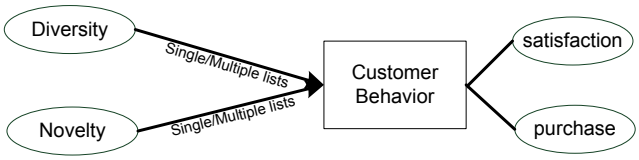

Figure 2: Instantiated model with dependent and independent variables.

To exemplify the usage of our model we designed the following experiment:

"A movie website with recommendations is presented to the experiment participants. First, the participants are asked to register and enter their movie preference. After that, each participant will obtain 3 *electronic vouchers* that can be use to buy 3 movies. Each voucher can be used to buy one movie. Next, the users are presented with a list of recommendations. Based on these recommendations, the participants will select movies for purchase. They can use all the vouchers at one time or save the vouchers for the future. After choosing the movies, we present a survey to the participants and ask if they are satisfied with the recommendation system."

We assume that the participants will carefully choose movies as they can postpone the choice and use the points for future movies in case they do not feel "tempted" by any of the recommended items. To present the recommendations, the user interface is also randomly selected. That means the recommendations can be presented only in a single list or in multiple lists that are used to separate basic, diverse and novel recommendations. In Table 1 we present a sketch of the factor design for the movie website experiment. The situations 1 to 4 in the table describe different configurations for the independent and moderator variables.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper4.pdf

Table 1. Factor design for the movie website experiment.

	Diversity	Novelty	User interface	
			Single List	Multiple Lists
1	without	without	Basic	Basic lists
2	with	without	0.3 diversity	Basic list
				Diverse list (1.0 diversity)
3	without	with	0.3 novelty	Basic list
				Novel list (1.0 novelty)
4	with	with	0.3 diversity, 0.3 novelty	Basic list
				Diverse list (1.0 diversity)
				Novel list (1.0 novelty)

The basic list contains the items selected by the recommendation algorithm. Novel lists are generated by manipulating the basic recommendation lists to include more novel items among the *top-n* items, considering that *n* is the number of items that will be presented to the user. In situations 2 to 4 in Table 1, a list with values 0.3 novelty stands for one list with novelty degree of 0.3, and a similar approach is used for diverse lists. This will be further explained in Section 4.

Diversity and novelty of recommendations can be designed in the form of binary or continuous. While the binary form is used to define the recommendations with or without diversity and novelty, continuous form defines diversity and novelty in the sense of various percentages. With the binary design, the recommendations can be configured as a between-subjects factor design. Thus, in each interface the result can be analyzed accordingly using a two-way ANOVA analysis [12]. When we employ the continuous design, the result can be analyzed using a regression analysis. The two designs can mutually confirm or supplement their findings. In addition, the experimental result of the effect of different interface designs can be analyzed based on A/B split testing. Based on this data, we can then analyze if and to what extent diverse or novel recommendations affect customer behavior and how to provide an appropriate interface when we introduce more diversity and novelty into the recommendations.

4. EVALUATING NOVEL AND DIVERSE RECOMMENDATIONS

In this section we focus on two quality factors: novelty and diversity. Novelty is related to items the user was not aware of. Diversity is generally defined as the opposite of similarity [17]. To implement these quality factors in an experiment, we should be able to control the degree of novelty and diversity in a recommendation list and multiple lists, considering the specific user interfaces involved in the experiment.

We propose to use a combination of three factors to identify novel items in a list of items: (1) "freshness" of an item (i.e., items that were recently launched), (2) non-popularity (popular items are not considered novel) and (3) limited relation to the long-term user profile (e.g. by previous ratings, feedback or views of this item in previous sessions). Each item is scored according to each factor in a scale from 0 (low) to 1 (high), and then the degree of novelty of the item is calculated simply by the average of the score for the three factors. Considering that i is an item and that $0 \leq fresh(i), nonpop(i), unknw(i) \leq 1$ represent respectively the degree of freshness, non-popularity and lack of relation to the user long term profile, the novelty $nov(i)$ of item i is defined by:

$$nov(i) = \frac{fresh(i) + nonpop(i) + unknw(i)}{3}$$

To calculate the degree of novelty of a recommendation list, we use the novelty degree of the list items. Assuming that an item is considered to be novel if its novelty degree is higher, for example, 60%, the novelty degree of a recommendation list L is the proportion of items from the list whose novelty degree surpasses this threshold.

$$nov(L) = \frac{|\{i \in L \,|nov(i) > 0.6\}|}{|L|}$$

For the multiple lists interface, we propose two lists: the basic list (as directly given by the recommendation algorithm) and a list consisting only of novel items, $nov(L2) = 1.0$. Another option is to have one list with a low degree of novelty (e.g. $(L1) = 0.2$) and another with a high degree of novelty (e.g. $nov(L2) = 0.6$).

The most explored method for measuring diversity uses item-item similarity. The diversity of a list of items can then be measured based on the sum, average, minimum or maximum distance between pairs of items [17, 19]. We adopt a slight modification of the approach from [20] and use the intra-list similarity metric (ILS). Considering that B is a set of items, this metric is based on a function of $c: B \times B \rightarrow [0,1]$ that is supposed to measure the similarity between two items according to a predefined criterion. Then we calculate the ILS as follows:

$$ILS(L) = \frac{\sum_{i_k \in L} \sum_{i_e \in L, i_k \neq i_e} c(i_k, i_e)}{2}$$

The selection of function c is dependent on the available content information for each item and can also be dependent on the user's preferences. The simplest option is to consider c as the degree of intersection of the items' properties (such as size, color, weight or genre). If we consider a function $prop: B \times P \rightarrow \{0,1\}$ where P is the set of available item properties, we can define c as:

$$c\,(i_k, i_e) := \frac{\sum_{p \in P} prop(i_k, p).\, prop(i_e, p)}{|P|}$$

We therefore can measure the diversity of a list by means of ILS and c. As high values of ILS denote low diversity, we take the inverse of ILS and define $DIV(L)$ as the degree of similarity of a list of items.

$$DIV(L) = \frac{1}{ILS(L)}$$

When designing diversity in multiple recommendation lists, we can use a threshold to discriminate a list with high similarity (e.g. $DIV(L) > 0.6$) or one with low similarity (e.g. $DIV(L) < 0.3$).

The measurements $nov(L)$ and $DIV(L)$ can be manipulated by changing some of the items in L. One strategy to increase $nov(L)$ is to substitute some items j from L by an equal number of items i not yet in L, such that $nov(i) > 0.6 > nov(j)$; considering that the threshold used to calculate $nov(L)$ is 0.6. In a similar way we can manipulate the diversity of a list L by replacing items that differ very little from other items already in L, i.e., items i for which $c(i,k)$ is high for several other elements k from the same

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper4.pdf

list L should be replaced by other items j, not yet in L, for which $c(j,k)$ is low for as many items k from L as possible.

5. CONCLUSION

Although many previous evaluations of recommender systems used accuracy as the only quality factor to be taken into account, recent works have shown that other metrics are also related to the user perception of quality of the recommendations. A further investigation of quality factors as diversity, novelty and serendipity lead us to conclude that the users' perception of these factors is highly linked to the system's interface.

This paper modeled the relations between evaluation metrics, the recommender interface and the customer behavior. Our main contribution is a first general model for experiments to evaluate recommender systems that aim to investigate the effects of recommendation quality factors and user interface on customer behavior. Besides proposing this general model, we also provided an example of how to instantiate the model for one specific experiment concerned with the evaluation of diversity, novelty and interface on customer purchase and satisfaction.

We consider that exploring other quality factors than accuracy is an important step towards improving the impact of recommenders and strongly believe that factoring in the user interface is crucial to realistically evaluate the users' perception of the quality of the recommendations. An interesting point is that the expected results could be used in line with the business strategy. For example, presume we found that with multi-list interface, diversity significantly affects customer purchase. Thus if we intend to promote certain product, we have more chances to advertise this product by including it in the diverse recommendations.

Although the designed experiment has not yet been completed, our study provided a first guideline on how to incorporate user interface aspects better in the recommender systems' evaluation. The experimental results are expected to show how the quality of recommendations could be optimized by presenting appropriate user interfaces. We intend to contribute to future research on how to examine the effects of evaluation metrics and the user interfaces in recommender systems so that further quality factors can be meaningfully evaluated.

6. REFERENCES

[1] Adomavicius, G. and Tuzhilin, A. 2005. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp. 734-749.

[2] Bodapati, A.V., Recommendation systems with purchase data. Journal of Marketing Research. 45(1). pp. 77-93.

[3] Campbell, D.T. and Stanley, J. (1963), Experimental and quasi-experimental designs for research. Houghton-Mifflin, Boston, Massachusetts, USA.

[4] Chen, L. and Pu, P. 2007. Preference-Based Organization Interfaces: Aiding User Critiques in Recommender Systems. Lecture Notes In Artificial Intelligence, vol. 4511. pp 77-86.

[5] Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. 1999. Combining collaborative filtering with personal agents for better recommendations. Conference of the American Association of Artificial Intelligence, Florida, USA. pp. 439-446.

[6] Gronroos, C. 1983. Strategic management and marketing in the service sector. Marketing Science Institute. USA.

[7] Hayes, C. Massa, P., Avesani, P., and Cunningham, P. 2002. An on-Line Evaluation Framework for Recommender Systems. Workshop on Personalization and Recommendation in E-Commerce, Malaga, Spain.

[8] Herlocker J., Konstan J., Terveen L. and Riedl J. 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1), pp. 5–53.

[9] Iaquinta, L., Gemmis, M., Lops, P. and Semeraro, G. 2008. Introducing serendipity in a content-based recommender system. 8th International Conference on Hybrid Intelligent Systems, Barcelona, Spain. pp 168-174.

[10] Jannach, D., Hegelich K.: 2009. A case study on the effectiveness of recommendations in the Mobile Internet, ACM Conference on Recommender Systems, New York, pp. 205-208.

[11] Jarvenpaa, S.L., Dickson, G.W. and DeSanctis, G. 1985 Methodological issues in experimental IS research: experiences and recommendations, MIS Quarterly, 9(2), pp.141-156.

[12] Juran, J.M., Gryna, F.M. and Bingham, R.S. 1974. Quality control handbook, 3rd edition, McGraw-Hill, New York, USA.

[13] Kamahara, J., Asakawa, T., Shimojo, S. and Miyahara, H. 2005. A community-based recommendation system to reveal unexpected interests. 11th International Multimedia Modeling Conference, Melbourne, Australia. pp. 433 – 438.

[14] Mcnee, S., Riedl, J and Konstan, J. 2006. Accurate is not always good: How Accuracy metrics have hurt recommender systems, Conference on Human Factors in Computing Systems, Quebec, Canada. pp. 1-5.

[15] Murakami, T., Mori, K. and Orihara, R. 2008. Metrics for evaluating the serendipity of recommendation lists. New frontiers in artificial intelligence, Lecture Notes in Computer Science, vol. 4914 pp. 40-46.

[16] Schafer, B., Konstan, J. and Riedl, J. 2001. E-Commerce Recommendation Applications, Journal of Data Mining Knowledge Discovery, 5(1-2), pp. 115-153.

[17] Shani, G. and Gunawardana, A. 2009. Evaluating Recommendation Systems. Microsoft research, Technical report, No. MSR-TR-2009-159.

[18] Senecal, S. and Nantel, J. 2004. The influence of online product recommendations on consumers' online choices. Journal of Retailing. 80(2), pp. 159-169.

[19] Zanker, M.; Bricman, M.; Gordea, S.; Jannach, D.; and Jessenitschnig, M. 2006. Persuasive online-selling in quality and taste domains. In Proceedings of 7th Intl. Conference E-Commerce and Web Technologies, Krakow, Poland, pp. 51–60.

[20] Ziegler, C., McNee, S. M., Konstan, J. A., and Lausen, G. 2005. Improving recommendation lists through topic diversification. In Proceedings of the 14th International World Wide Web Conference, Chiba, Japan , pp. 22-32.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

Information Overload and Usage of Recommendations

Muhammad Aljukhadar
HEC Montreal
3000 Cote-St-Catherine
Montreal, Canada H3T2A7
1-514-340-7012
Muhammad.aljukhadar@hec.ca

Sylvain Senecal
HEC Montreal
3000 Cote-St-Catherine
Montreal, CanadaH3T2A7
1-514-340-6980
Sylvain.senecal@hec.ca

Charles-Etienne Daoust
Cossette Communication
Canada
1-514-340-6980
charles-etienne.daoust@hec.ca

ABSTRACT
This research examines the antecedents of information overload and recommendation agents' consultation and their effects on reactance and choice quality. We propose that information overload and the user need for cognition affect the tendency to employ decision heuristic (consulting a recommendation agent) and shape the user reactance to recommendations. A fully randomized experiment with different levels of information loads that involved 466 individuals with the task of choosing a laptop and the option to consult a recommendation agent is performed. Results show that users opted to consult the recommendation agent more as information loads and as perceived overload increases and that product recommendations were salient in enhancing choice, particularly when the information was less diagnostic (for choice sets with proportional distribution of attribute levels across alternatives). Results further reveal that as perceived overload increases, people show less reactance to recommendations. Whereas users consulting the recommendations at higher overload levels had generally better choices, they showed higher confidence in their choices only when they conform rather than react to recommendations.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Agents and Web-services.

General Terms
Management, Measurement, Human Factors, Performance, Design, Theory.

Keywords
Recommendation Agents, Information Overload Theory, Reactance Theory.

1. INTRODUCTION
When making purchase decisions, users typically process large amounts of information. As people shop online to save time and effort, retailers are required to effectively manage product information delivered on their e-stores. The many choice possibilities associated with large choice sets represents an opportunity and challenge for consumers and retailers [7, 9]. To help customers reduce the cognitive effort while enhancing their decision, retailers incorporate on their e-stores agents that filter, optimize, and organize product information. Product recommendations are decision-aid tools that support rather than replace consumer decision-making by suggesting one or more product that closely matches consumer preferences [26]. In effect, decision support systems are heuristics that partly alleviate processing effort while maintaining an acceptable level of choice accuracy [10]. Xiao and Benbasat [28 p. 137] recently provide an extensive review of the RA literature, and conclude that "by providing product recommendations based on consumers' preferences, RAs have the potential to support and improve the quality of the decisions consumers make when searching for and selecting products online as well as to reduce the information overload facing consumers and the complexity of online searches." This explains why 40% of retailers plan to integrate some personalized recommendations on their e-stores [6].

While research studied various designs of recommendation agents, it has not investigated the factors triggering consumers to consult the recommendations nor the cases where product recommendations are vital to choice enhancement [10, 27, 28]. Indeed, research is yet to assess the factors that lessen the user reactance to recommendations [7]. Lurie [18 p. 484] indicates that "… in the age of the Internet, developing an understanding of how information-rich environments affect consumer decision making is of crucial importance. Given the disparate ways in which product information can be presented to consumers and the high potential for information overload in online environments, it is important to use measures that capture the multiple dimensions of information."

The contribution of this article is four-fold. First, the article examines the relation between the delivered information load in the choice set and perceived overload by simultaneously manipulating the number of alternatives, number of attributes, and the distribution of attribute levels across the alternatives. Second, it assesses the role of information overload on employing decision heuristics (the tendency to consult the recommendation agent) while considering the role of need for cognition. Third, it investigates how information overload and need for cognition shape users' reactance to recommendations. Fourth, it examines the impact on choice quality and confidence. We next briefly review the literature and present the study conceptual framework. The methodology section reports the details of the pretest and the experiment. Results are then presented. The paper concludes with a summary of findings and implications on theory and practice.

2. CONCEPTUAL FRAMEWORK
Research showed the effects of information overload on the choice and purchase of different products: Laundry detergent [13], rice and prepared dinner [14], peanut butter [25], houses [19], calculators [18], and CD players [17]. Research indicates that variations in the amount of information impact the decision processes, which affects decision quality. Information overload

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

happens because of humans' limits in assimilating and processing information within any timeframe [13, 19]. When consumers are faced with high levels of information, their limited capacity to process information becomes overloaded, which results in dysfunctional consequences such as cognitive fatigue and confusion [8, 16, 20, 21, 25].

Several measures were used to capture the amount of product information. Researchers have traditionally manipulated the alternative and attribute levels in product choice sets [13, 19]. While this line of research has made substantial contribution, discrepancies were noted [12, 19, 20, 21]. More recently, the concept of information structure was introduced and shown to have a role in determining overload; this concept asserts that when measuring information loads, both the number and probability of outcomes should be considered (for a discussion, see [18]). When the distribution of attribute levels for instance is proportional across the alternatives (e.g., half the laptops in a given choice set are equipped with Intel and half with AMD processors), information load will be higher than for a disproportional distribution (e.g., 3/4 with Intel and 1/4 with AMD processors). This is because a disproportional distribution increases information diagnosticity [18]. Information load in a choice set can hence be affected by the number of alternatives, number of attributes, as well as the distribution of attribute levels across the alternatives (attribute distribution hereafter) [17, 18]. One purpose of this research is to manipulate these three dimensions over a range that is wider than prior work and to assess the impact on perceived overload and choice. After information-processing capacity is surpassed, information increments were found to lead to modest or insignificant reductions in decision quality [8, 18]. As research stipulates a complex rather than a linear relation between information load and perceived overload [8, 14], we expect a nonlinear relation to better describe the relation between these two factors (P1).

It is plausible to assume that under high overload levels, consumers do use heuristics to maintain the cognitive effort at acceptable levels. Indeed, consumers adapt decision strategy according to product information, task, and environment [5, 23]. In complex choice situations, consumers for instance become more selective in acquiring and processing information [23]. Because consulting product recommendations can be seen as information-processing heuristic [10, 27, 28], we theorize that the utility of consulting product recommendations increases with information overload. Under high overload levels, consumers behave as satisficers (vs. optimizers) and thus use more an information-processing reduction strategy [19]. Therefore, we expect that (P2) consumers will tend to consult the recommendations more as (a) information load increases and as (b) perceived overload increases. Figure 1 depicts the study conceptual framework.

Consumers have divergent needs for information. Need for cognition (the consumer tendency to engage in effortful thinking) was cited as an important factor of attitudinal and behavioral change [4]. Consumers low on the need for cognition tend to avoid activities requiring high cognitive effort and to engage in heuristic strategies [11]. We thus expect need for cognition to attenuate the tendency to consult the recommendations such that as information overload increases, the lower the need for cognition is, the more the consumer will consult product recommendations (P3).

Consumers do react to product recommendations because they limit their choice freedom [7]. Under high overload levels, consumers behave as satisficers as opposed to optimizers [19]. Because consumers are adaptive decision makers [3], we propose that the higher the information overload becomes, the more the consumer will conform to recommendations (P4). This proposition finds support in the self-regulation research; information overload can be seen as a resource depletion mechanism that "enhances the role of intuitive reasoning by impairing deliberate, careful processing" of information [24, p. 344]. Need for cognition is also expected to shape reactance so that under higher levels of overload, the lower the need for cognition is, the less the consumer will react to product recommendations (P5).

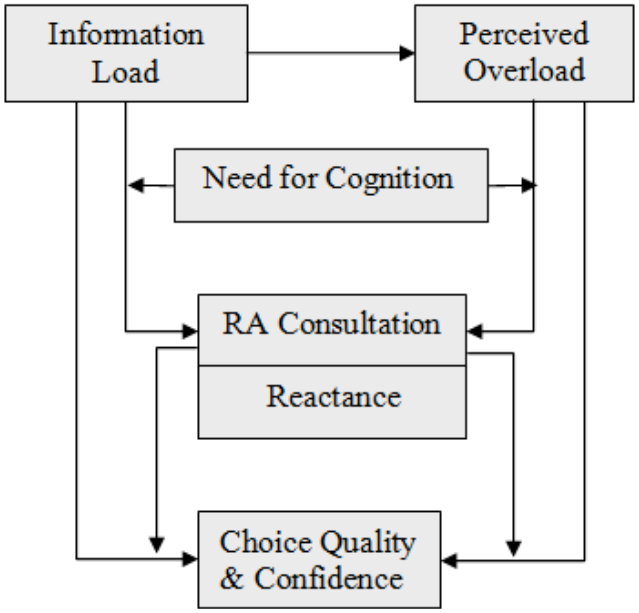

Figure 1. Research Framework.

We finally study the impact of information overload and product recommendations on choice quality and confidence. Theory posits a salient role for recommendations on choice quality in complex choice situations [27]. In effect, choice quality suffers when the processing effort exceeds processing limits [23]. As product recommendations help consumers improve choice by concentrating on the alternatives that best match their preferences [10], product recommendations should uphold choice quality as information overload increases (P6) [3, 10, 15, 19, 28]. Because the negative role of information overload on choice is prominent in the case of a proportional versus disproportional attribute distribution [18], we theorize that the impact of product recommendations on choice quality will be particularly salient for choice sets with proportional attribute distribution (P7). According to Fitzsimon and Lehmann [7], recommendations reduce uncertainty for consumers who do not react to recommendations. We hence expect that consumers who consult and conform to product recommendations will have higher choice confidence than consumers who consult but react to recommendations (P8).

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

3. METHODOLOGY

3.1 The Experimental Site and the Recommender System

An e-store was created for "Portable Direct" using professional Web design service; a fictitious retailer name was used to control for retailer preferences [1]. The computer laptop was chosen as product category because (a) it is a complex product thus consumers are expected to be attentive during choice, (b) it has many known attributes, which allows a meaningful manipulation at high number of attributes, (c) it is a search product (attributes can be communicated using the Web), and (d) it is a product that consumers shop for online, which improve the ecological validity. Though pretested (see the Appendix), manipulation levels were adapted from the literature. Three levels of alternatives (6, 18, and 30) were chosen because research investigating this factor along with attribute distribution considers only two alternative levels (18 and 27 in [17, 18]) and because little research manipulated for choice sets with low alternative level [19]. Three levels of attributes (15, 25, and 35) were chosen because research investigating this factor along with attribute distribution considers only two attribute levels (9 and 18 in [17]). Whereas few studies manipulated for 20 attributes or more [8, 19], including higher number of attributes is necessary as consumers consider many attributes when shopping for complex products. Akin to prior work [17, 18], the distribution of attribute levels across the alternatives had two levels (proportional vs. disproportional distribution); the attributes provided in a choice set were manipulated according to one of these levels.

The participant rates the importance (weight; 1-7) of each of the 35 attributes (this step is performed before the participant is randomly assigned to one of the eighteen experimental conditions). Then, the score of each potential choice (each laptop in the choice set provided under a particular condition) can be determined by the following formula (Weighted Additive Rule; Payne, Bettman, and Johnson 1993):

$$S_{jk} = \sum V_{ij} P_{ik}$$

Where: S = Global score of alternative j for consumer k.

i = Attribute;

j = Alternative (laptop);

k = Consumer;

P = Weight of attribute i for consumer k;

V = A priori value of attribute i applied by system and associated with alternative j.

That is, the WADD determines the score of a given alternative j (for consumer k) by multiplying the weight of each attribute (provided by consumer k) by its a priori value, and then adding the obtained values of all attributes. The alternative with the highest score (i.e. the one that optimizes consumer k's utility function) is then suggested by the recommendation agent (should consumer k choose to consult the agent by clicking the link provided).

3.2 Pretest and Measure

Each participant had to choose a laptop with the option to consult the recommendations (between-subject design). Recommendations consultation and if consulted whether the recommended product was chosen are observed variables. Perceived overload was measured using two seven-point items (There was too much information to make a choice; I wanted to receive more information about the different products before making my choice). Similar to [13, 19], choice confidence was measured using three items (I am confident that I made the best possible choice based on my needs; I am satisfied with the choice I made; I am certain that I made a good choice; α=0.93). Need for cognition was measured using the 18-item scale ([4], α=0.82). As decision makers draw on their experience and knowledge of product category, product experience (three-item from [22], α=0.95) and product category involvement (four items adapted from [2], α=0.92) were measured and controlled for. See the Appendix for details of the pretest and manipulation checks.

3.3 Stimuli

Participants were informed that their task consisted of choosing a laptop as they would in an actual purchasing situation. The task page described "Portable Direct" as a well-established online retailer of product category and asked the participants to navigate its e-store (made available through a link provided after the participants entered personal attribute preferences) to choose the "The laptop you would seriously consider buying". Participants were told to take as much time as needed and to freely consult the information available on the website. A time constraint was not imposed because this would be inconsistent with real-life situations and because this would result in eliminating a portion of participants based on some cut-off value. In effect, time pressure was shown to influence information overload [8]. Before a participant was randomly assigned to one of the eighteen conditions, a second page asked the participant to rate the importance of each attribute (to estimate the participant utility function so that the recommendation agent could suggest the optimal choice; Weighted Additive Rule WADD as in [23]). Depending on the assigned condition, the e-store provided the participant with a finite choice set (e.g., six alternatives each with fifteen attributes for conditions one and two in the Appendix). Similar to factual e-stores, each alternative appeared in a tabular format with the attributes headed by the laptop photograph. The alternatives that made the choice set were presented on the same page. To avoid presentation bias, the order of alternatives was randomized for each participant in a given condition. Brand was concealed to reduce the possibility of following a brand heuristic and to entice participants to make choice using the information provided. This is akin to prior work [17]. Participants had the option to consult the recommendations by clicking on a hyper link labeled "Click here for our recommendation according to your preferences" located at top of the choice set provided. After making their choice, participants were presented with the measure items.

3.4 Sample

An invitation to participate in a "Study on e-commerce" was sent to consumers randomly chosen from a large consumer panel belonging to a North American market research company. Of the 472 responses received, 466 were complete and retained. Sample demographics distribution (see the Appendix) shows that the

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

sample was well distributed across consumer population with no important bias toward a particular segment.

4. RESULTS

A comprehensive analysis of the data with a path model was not performed because it was not feasible (i.e., central variables in the model such as RA consultation and reactance to recommendation were binary; in addition, an important exogenous variable-information load-is ordinal and reflected by one item). As such, ANOVA and regression analysis were used in testing the propositions (except for P2 through P5 where logistical regression were used because the dependent variable was binary).

The main effect for information load (called interchangeably information bits; [17, 18], see the Appendix section) on perceived overload was significant (F=23.88, p<0.001); this result stays reliable when controlling for product involvement and experience (only product experience was significant covariate; B= -0.085, F=5.34, p=0.021). A curvilinear quadratic curve solution explained more variance (R^2=0.264) in the relationship between information load and perceived overload than a linear (R^2=0.224) or a logarithmic (R^2=0.248) solution (Figure 2).

Binary logistical regression was performed to test the impact of information load on recommendations consultation as well as the attenuating role of need for cognition. Information loads increment led to more recommendation consultation by means of main effect (B=0.164, Wald=6.00, p<0.05). In addition, the interaction between information loads and need for cognition was significant in the predicted direction (B=-0.031, Wald=5.587, p<0.05). Similarly, logistical regression was performed to test the impact of perceived overload on recommendations consultation and the attenuating role of need for cognition. Perceived overload did lead to more consultation of recommendations (B=0.344, Wald=4.06, p=0.044) and the interaction between perceived overload and need for cognition was significant in the predicted direction (B=-0.077, Wald=5.71, p=0.017). The direct effects of the alternative, attribute, and attribute distribution levels and their interactions on recommendations consultation were examined and showed insignificance (all p's>0.10 NS).

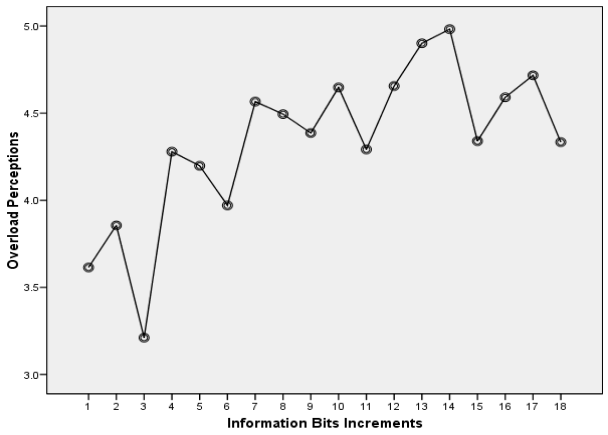

Figure 2. Information load effect on perceived overload.

To test the impact of perceived overload and need for cognition on reactance, we applied binary logistical regression on the observations that consulted the recommendations (n=178). As expected, perceived overload was significant factor in predicting the conformation (vs. reactance) to recommendations (B=0.91, Wald=8.10, p=0.004). In addition, the interaction between perceived overload and need for cognition was significant (B= -0.131, Wald=4.52, p=0.034), which shows that as perceived overload increases, the lower the consumer was on need for cognition, the less reactance to recommendations the consumer would exhibit. Alternatively, neither information load nor its interaction with need for cognition were significant in predicting reactance (all p's>.34 NS). We further tested the direct impact of the levels of alternatives, attributes, and attribute distribution on reactance and found no significant effects (all p's>.31). These results collectively show that perceived overload, rather than information loads, was the determinant factor in predicting reactance to recommendations.

Choice quality was measured by the distance between the participant actual and optimal choice (Weighted Additive Rule WADD; [23]). This is akin to past work [13, 16, 19]. The expected interaction between information load and recommendations consultation was significant (F=1.68, p=0.012; Figure 3 Up). Similarly, we found support to the proposition that recommendations consultation upholds choice quality as perceived overload increases because the interaction between perceived overload and recommendations consultation was significant (F=1.61, p=0.036; Figure 3 down).

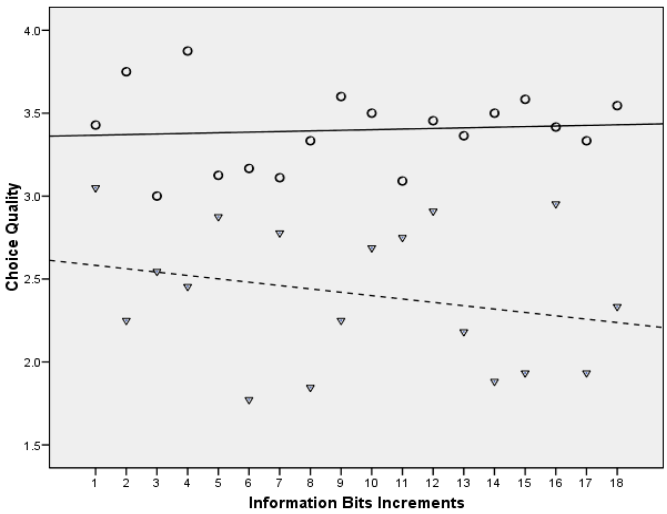

Figure 3a. Recommendations effect on choice quality (upper line: RA consulted).

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

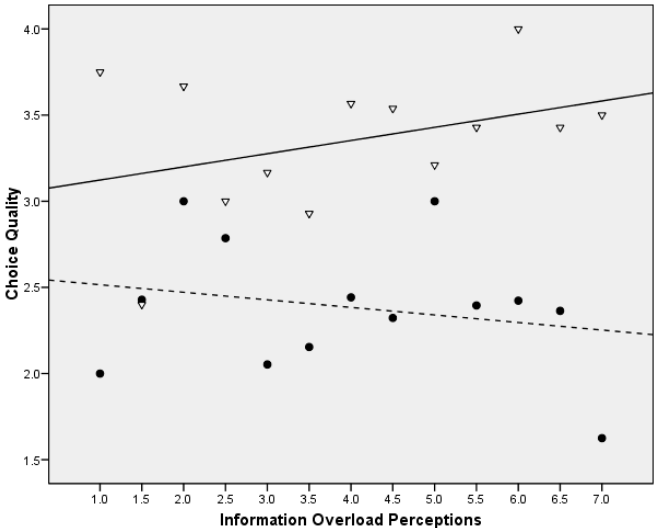

Figure 3b. Recommendations effect on choice quality (upper line: RA consulted).

We then tested the proposition that product recommendations effect on choice quality is salient for choice sets with proportional attribute distribution (P7). We found support to this proposition by means of a three way interaction (Number of Attributes x attribute distribution x recommendations consultation; F=2.47, $p<0.05$; Figure 4). This interaction shows the recommendations to enhance choice quality for choice sets with proportional distribution of attribute levels across the alternatives at all attribute levels (Appendix for means). The interaction also highlights that recommendations consultation improved choice for all choice sets only when the number of attributes became high. We finally tested and found support to the proposition that consumers consulting and conforming to recommendations will have higher choice confidence than consumers consulting and reacting to recommendations (5.13 vs. 4.41, F=8.55, p=.004).

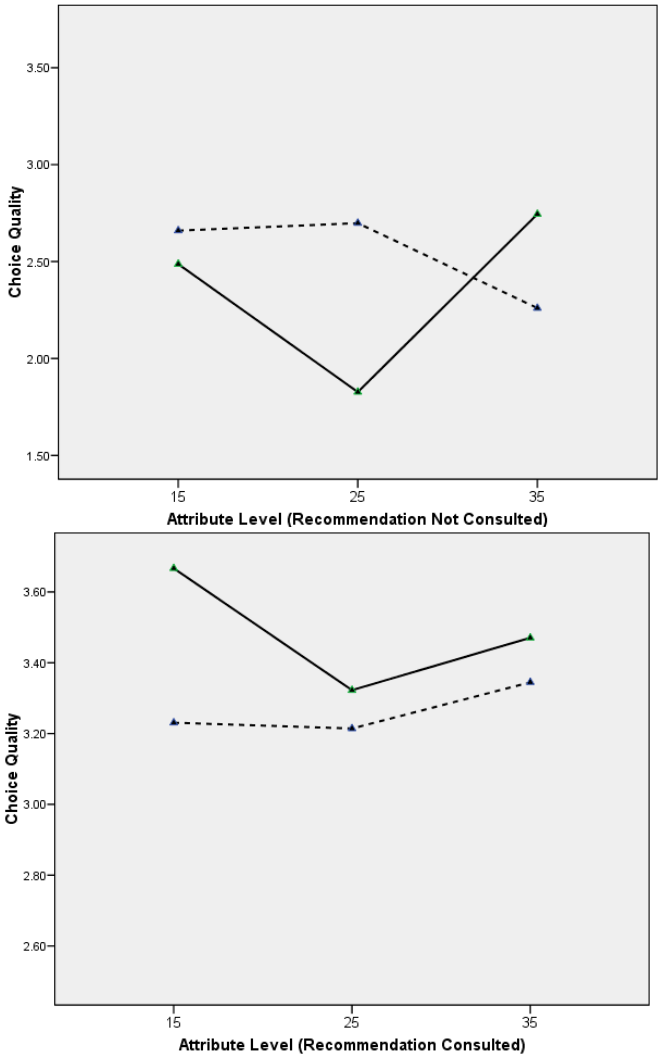

Figure 4. Recommendations effect on choice quality for choice sets with proportional versus disproportional distribution of attribute levels across the alternatives.

— = Proportional attribute distribution.

- - - = Disproportional attribute distribution.

5. DISCUSSION

The experimental results lend support to research propositions. Results suggest a curvilinear relation between information load and perceived overload, which indicates that the impact of additional increments in product information after some levels (condition 7 shown in the Appendix) are not as influential in driving overload perceptions. The consumer use of decision heuristics at high levels of information overload helps explaining this finding. Findings lend support to the notion that the utility of consulting product recommendations increases as the information load and as perceived overload increases. Consumers did use an information-processing heuristic by consulting product recommendations more as information overload increases. Moreover, this tendency was higher for consumers low on the need for cognition. Importantly, consumers appear to conform (vs. react) to recommendations more at high levels of perceived overload. Further, the lower the need for cognition was, the less the consumer reacted to recommendations at higher levels of information overload.

The findings show the positive effects of product recommendations on choice quality at high levels of information loads and overload perceptions. The positive impact of recommendations on choice quality was particularly salient for choice sets with proportional distribution of attribute levels across the alternatives. Finally, choice confidence improved for consumers who consulted and conformed (vs. reacted) to recommendations. In effect, the recommendations might have made the accuracy feedback as immediate and tangible as the effort feedback by signaling to consumers that a product in the choice set is more optimal than the initially considered one [5], which might have triggered consumers to have lower levels of confidence in their choice if they reacted to the recommendations.

This research contributes to theory by studying the relation between information loads and overload perceptions over a wide range for three factors deemed to determine the information load

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

and by showing that consumers indeed do employ decision heuristics in response to information overload. People appear to regard the use of product recommendation agent as information-processing reduction heuristic. This research further established a link between information overload and reactance to recommendations and underlined the role of need for cognition. It contributes to the recommendation agents' literature by showing the impact of recommendations on choice at different information overload levels and by showing the salient effect of recommendations on choice quality for sets with proportional distribution of attribute levels across the alternatives.

Several practical implications emerge. Integrating a recommendation agent based on consumer preferences appears to be beneficial for consumers and retailers (by helping consumers make quality choices at high levels of information overload). Recommendations enhance choice, particularly as information load and perceived overload increases. In addition, recommendation agents appear to have particular influence on choice when product information is less diagnostic (attribute levels are proportionally distributed across the alternatives in the choice set). Finally, the outcome of recommendation agents can be optimized as consumers in general show less reactance to recommendations at higher levels of information overload.

This work has limitations. Although the study sample comprised actual consumers randomly selected from large consumer panel, the sample was self-selected. Nonetheless, the sample distribution across the consumer population was satisfactory. The research considered only one product category and did not examine whether similar effects are obtainable for less complex and for experience products. Further, this research did not investigate the effects of information overload and product recommendations on shopping enjoyment and long term performance measures such as consumer loyalty and retention. These topics are potential extensions to this line of research.

6. APPENDIX

6.1 Experimental conditions (Information Load*)

Information load Increment (condition)	*Attribute Levels Distribution*	*Number of Alternatives*	*Number of Attributes*	*Participants in condition (% of total)*
1	Disproportional	6	15	27(5.8)
2	Proportional	6	15	20(4.3)
3	Disproportional	18	15	25(5.4)
4	Proportional	18	15	19(4.1)
5	Disproportional	6	25	32(7.3)
6	Proportional	6	25	34(7.3)
7	Disproportional	18	25	27(5.8)
8	Proportional	18	25	22(4.7)
9	Disproportional	30	15	21(4.5)
10	Proportional	30	15	30(6.4)
11	Disproportional	6	35	31(6.7)
12	Proportional	6	35	22(4.7)
13	Disproportional	30	25	22(4.7)
14	Proportional	30	25	27(5.8)
15	Disproportional	18	35	27(5.8)
16	Proportional	18	35	33(7.1)
17	Disproportional	30	35	21(4.5)
18	Proportional	30	35	26(5.6)

* Information load increments are determined following Lee and Lee (2004) and Lurie (2004); also fulfilling the approximation: Information Load=No. of Alternatives + 2(No. of Attributes)

6.2 Pretest and Manipulation Checks

A pretest was performed to ensure task and measure comprehensibility [8], to check the manipulation of independent variables and to inspect the distribution of control variables. The pretest ensured that an increment from six (and eighteen) to thirty alternatives resulted in a noticeable change in information load. The pretest included three sections: The first contained the manipulation checks, the second examined product experience level and where the product category was relevant for the participant pool (e.g., manipulating the attributes level would be realistic and meaningful). The third section helped determining the 35 most important attributes (of 45 attributes identified using two retailing websites) to be included in experiment (each attribute was evaluated using a Very Important/Not Important at All seven-point item).

Six questionnaire versions were created for the pretest, all sharing the items of product experience and involvement, as well as attribute importance evaluation (the versions differed only in the first section). The first two versions were developed to check the manipulation of number of alternatives (6, 18, and 30). The two versions differed in the order the three levels were presented to each participant (i.e., while the order was 6-18-30 in the first version, the order was reversed in second version). This eliminated the possibility that a respondent rated level one as having fewer alternatives than levels two and three because it was displayed first. Similar steps were taken in versions three and four, which checked the manipulation for number of attributes. Versions five and six examined the manipulation for attribute distribution (proportional vs. disproportional). Version five (six) assessed the manipulation for a proportional (disproportional) distribution of attribute levels across the alternatives (both for the price attribute).

An invitation to participate in the pretest was emailed to 116 consumers (convenience sample). 77 useable responses were received. Because the measure (for both the alternatives level and attributes level) was within-subjects, ANOVA with repeated measures was used to analyze the input. For attribute distribution, a chi-square test was used. The 32 participants that evaluated alternatives level had to respond to a seven-point bipolar item (What do you think of the quantity of laptops offered: Not enough to make a choice/too much to make a choice) (item repeated for each of the three levels presented to the respondent).

The analysis showed that participants perceived significantly different information loads between each of the three levels (M6=2.66, M18=4.81, M30=4.94; F6-18 (1, 31)=69.65, F6-30(1, 31)=139.7, F18-30(1, 31) =27.59, all p-values<0.001). Similarly, the 23 participants evaluating the attributes level had to respond to the seven-point bipolar item (What do you think of the quantity of attributes offered: Not enough to make a choice/too much to make a choice; item was repeated for each of the three levels presented to the participant). The analysis showed that participants reported significantly different information loads between each of the three levels (M15=2.87, M25=4.30, M35=4.87; F15-25(1, 22)=77.85, F25-35(1, 22)=10.33, F15-35(1, 22)= 97.32, all p-values<0.01). The 22 participants evaluating the success of attribute distribution manipulation responded to a binary item (Was the number of laptops priced at $600 different or similar to the number of laptops priced at $750 and $900?). For (dis)proportional structure, the number was (not) equal. Participants in the (dis)proportional structure condition reported (un)equal distribution of the price attribute across alternatives ((1, 22) = 12.32, $p < 0.01$).

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

The second section (shared for all participants) showed that the laptop computer is a product bought and used frequently by participants (87 percent of participants indicated using or to have used a laptop regularly; 75 percent of participants have already bought a laptop). This section also showed the internal consistency for product experience items (α=0.96) and product involvement items (α=0.87) and clarified the sample distribution according to these variables.

Attributes were assigned to experimental conditions using the pretest input. Attributes that have higher weights appeared more often in conditions with fewer attributes. This was done because the inclusion of an attribute in a choice set renders the attribute more important for the decision maker [9]. Consequently, including less important attributes in a choice set made up of few attributes would inflate the attribute's importance. In effect, choice sets containing only less relevant attributes for the alternative (choice sets that do not provide basic and important attributes such as price, processing speed, or memory size) are unrealistic and would reduce ecological validity.

6.3 Sample Demographics (n=466; 56.9% females)

Age: 11.6% ages 18-24, 26.4% 25-34, 20.0% 35-44, 19.7% 45-54, 9.9% 55-64, 12.4% 65+. Education level: 19.6% Primary/secondary education level, 70.8% Undergraduate degree, 9.7% Graduate degree. Income: 14.2% less than $15K, 18.9% 15-29K, 29.0% 30-44K, 19.7% 45-59K, 9.7% 60-74K, 7.5% 75K or higher. Marital status: 28.8% single, 57.9% married/common law partner, 13.3 other status. Employment: 9.5% students, 78.6% working full-time, 7.1% working part-time, 4% searching.

6.4 Choice Quality Means

Attribute Level	*Recommendation Consulted*	*Attribute Distribution*	*M*	*SE*	*95% Confidence (Lower/Upper)*	
15	No	Disproportional	2.660	.154	2.357	2.963
		Proportional	*2.487*	*.169*	*2.155*	*2.820*
	Yes	Disproportional	3.231	.207	2.823	3.638
		Proportional	*3.667*	*.193*	*3.287*	*4.046*
25	No	Disproportional	2.698	.145	2.413	2.983
		Proportional	*1.827*	*.147*	*1.539*	*2.115*
	Yes	Disproportional	3.214	.200	2.822	3.607
		Proportional	*3.323*	*.190*	*2.949*	*3.696*
35	No	Disproportional	2.260	.149	1.966	2.554
		Proportional	*2.745*	*.154*	*2.442*	*3.048*
	Yes	Disproportional	3.345	.196	2.959	3.731
		Proportional	*3.471*	*.181*	*3.114*	*3.827*

7. REFERENCES

[1] Aksoy, L., Bloom, P., Lurie, N., and Cooil, B. 2006. Should Recommendation Agents Think Like People? *Journal of Service Research*, 8, 4, 297-315.

[2] Beatty, S. E. and Talpade, S. 1994. Adolescent Influence in Family Decision Making: A Replication with Extension. *Journal of Consumer Report*, 21, 2, 332-342.

[3] Bettman, J. R., Johnson, E. J., and Payne, J. W. 1990. A Componential Analysis of Cognitive Effort in Choice. *Organizational Behavior and Human Decision Processes*, 45, 1, 111-140.

[4] Cacioppo, J., Petty, R., and Kao, C.F. 1984. The efficient assessment of need for cognition. Journal of Personality Assessment, 48, 1, 306–307.

[5] Einhorn, H. J. and Hogarth, R. M. 1981. Behavioral Decision Theory: Processes of Judgment and Choice. *Journal of Accounting Research*, 19, 1, 1-31.

[6] eMarketer. 2008. Retailers Take Note: Video Sells! DOI =http://www.emarketer.com/Article.aspx?id=1006883

[7] Fitzsimon, G. J. and Lehmann, D.R. 2004. Reactance to Recommendations: When Unsolicited Advice Yields Contrary Responses. *Marketing Science*, 23, 1, 82-94.

[8] Hahn, M., Lawson, R., and Lee, Y. 1992. The Effects of Time Pressure and Information Load on Decision Quality. *Psychology & Marketing*, 9, 5, 365-379.

[9] Häubl, G. and Murray, K.B. 2003. Preference Construction and Persistence in Digital Marketplaces: The Role of Electronic Recommendation Agents. *Journal of Consumer Psychology*, 13, 3, 75-91.

[10] Häubl, G., and Trifts, V. 2000. Consumer Decision Making in Online Shopping Environments: The Effect of Interactive Decision Aids. *Marketing Science*, 19, 1, 4-21.

[11] Haugtvedt, C. and Petty, R. 1992. Personality and Persuasion: Need for Cognition Moderates the Persistence and Resistance of Attitude Changes. *Journal of Personality and Social Psychology*, 63, 2, 308-319.

[12] Jacoby, J. 1984. Perspectives on Information Overload. *Journal of Consumer Research*, 10, 4, 432-436.

[13] Jacoby, J., Speller, D., and Berning, C. 1974. Brand Choice Behavior as a Function of Information Load. *Journal of Marketing Research*, 11, 1, 63-69.

[14] Jacoby, J., Speller, D., and Berning, C. 1974. Brand Choice Behavior as a Function of Information Load: Replication and Extension. *Journal of Consumer Research*, 1, 1, 33-42.

[15] Johnson, E. J. and Payne, J.W. 1985. Effort and Accuracy in Choice. *Management Science*, 31, 4, 394-414.

[16] Keller, K.L. and Staelin, R. 1987. Effects of Quality and Quantity of Information on Decision Effectiveness. *Journal of Consumer Research*, 14, 2, 200-213.

[17] Lee, B.K. and Lee, W.N. 2004. The Effect of Information Overload on Consumer Choice Quality in an On-Line Experiment. *Psychology & Marketing*, 21, 3, 159-181.

[18] Lurie, N. H. 2004. Decision Making in Information-Rich Environments: The Role of Information Structure. *Journal of Consumer Research*, 30, 4, 473-486.

[19] Malhotra, N. K. 1982. Information Load and Consumer Decision Making. *Journal of Consumer Research*, 8, 4, 419-430.

[20] Malhotra, N. K. 1984. Reflections on the Information Overload Paradigm in Consumer Decision Making. *Journal of Consumer Research*, 10, 4, 436-440.

[21] Malhotra, N. K. 1984. Information and Sensory Overload. *Psychology & Marketing*, 1, 3&4, 9-21.

[22] Oliver, R. L. and Bearden, W. O. 1985. Crossover Effects in the Theory of Reasoned Action: A Moderating Influence Attempt. *Journal of Consumer Research*, 12, 3, 324-340.

[23] Payne, J. W., Bettman J. R., and Johnson, E. J. 1993. *The Adaptive Decision Maker*. New York, Cambridge University.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper5.pdf

[24] Pocheptsova, A., Amir, O., Dhar, R., & Baumeister, R. F. 2009. Deciding Without Resources: Resource Depletion and Choice in Context. *Journal of Marketing Research*, 46, 3, 344-356.

[25] Scammon, D. L. 1977. Information Load and Consumers. *Journal of Consumer Research*, 4, 3, 148-155.

[26] Senecal, S. and Nantel, J. 2004. The Influence of Online Product Recommendations on Consumers' Online Choices. *Journal of Retailing*, 80, 2, 159-169.

[27] Swaminathan, V. 2003. The Impact of Recommendation Agents on Consumer Evaluation and Choice: The Moderating Role of Category Risk, Product Complexity, and Consumer Knowledge. *Journal of Consumer Psychology*, 13, 1&2, 93-101

[28] Xiao, B. and Benbasat, I. 2007. E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Quarterly*, 31, 1, 137-209.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

A Novel Multidimensional Framework for Evaluating Recommender Systems

Artus Krohn-Grimberghe
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
artus@ismll.de

Alexandros Nanopoulos
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
nanopoulos@ismll.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
schmidt-thieme@ismll.de

ABSTRACT

The popularity of recommender systems has led to a large variety of their application. This, however, makes their evaluation a challenging problem, because different and often contrasting criteria are established, such as accuracy, robustness, and scalability. In related research, usually only condensed numeric scores such as RMSE or AUC or F-measure are used for evaluation of an algorithm on a given data set. It is obvious that these scores are insufficient to measure user satisfaction.

Focussing on the requirements of business and research users, this work proposes a novel, extensible framework for the evaluation of recommender systems. In order to ease user-driven analysis we have chosen a multidimensional approach. The research framework advocates interactive visual analysis, which allows easy refining and reshaping of queries. Integrated actions such as drill-down or slice/dice, enable the user to assess the performance of recommendations in terms of business criteria such as increase in revenue, accuracy, prediction error, coverage and more.

The ability of the proposed framework to comprise an effective way for evaluating recommender systems in a business-user-centric way is shown by experimental results using a research prototype.

Keywords

Recommender Systems, Recommendation, Multidimensional Analysis, OLAP, Exploratory Data Analysis, Performance Analysis, Data Warehouse

1. INTRODUCTION

The popularity of recommender systems has resulted in a large variety of their applications, ranging from presenting personalized web-search results over identifying preferred multimedia content (movies, songs) to discovering friends in social networking sites. This broad range of applications, however, makes the evaluation of recommender systems a challenging problem. The reason is the different and often contrasting criteria that are being involved in real-world applications of recommender systems, such as their accuracy, robustness, and scalability.

The vast majority of related research usually evaluates recommender system algorithms with condensed numeric scores: root mean square error (RMSE) or mean absolute error (MAE) for rating prediction, or measures usually stemming from information retrieval such as precision/recall or F-measure for item prediction. Evidently, although such measures can indicate the performance of algorithms regarding some perspectives of recommender systems' applications, they are insufficient to cover the whole spectrum of aspects involved in most real-world applications. As an alternative approach towards characterizing user experience as a whole, several studies employ user-based evaluations. These studies, though, are usually rather costly, difficult in design and implementation.

More importantly, when recommender systems are deployed in real-world applications, notably e-commerce, their evaluation should be done by business analysts and not necessarily by recommender-system researchers. Thus, the evaluation should be flexible on testing recommender algorithms according to business analysts' needs using interactive queries and parameters. What is, therefore, required is to provide support for evaluation of recommender systems' performance based on popular online analytical processing (OLAP) operations. Combined with support for visual analysis, actions such as drill-down or slice/dice, allow assessment of the performance of recommendations in terms of business objectives. For instance, business analysts may want to examine various performance measures at different levels (e.g., hierarchies in categories of recommended products), detect trends in time (e.g., elevation of average product rating following a change in the user interface), or segment the customers and identify the recommendation quality with respect to each customer group. Furthermore, the interactive and visual nature of this process allows easy adaptation of the queries according to insights already gained.

In this paper, we propose a novel approach to the evaluation of recommender systems. Based on the aforementioned motivation factors, the proposed methodology builds on multidimensional analysis, allowing the consideration of various aspects important for judging the quality of a recommender system in terms of real-world applications. We describe a way for designing and developing the proposed extensible multidimensional framework, and provide insights into its applications. This enables integration, combination and comparison of both, the presented and additional, measures (metrics).

To assess the benefits of the proposed framework, we have implemented a research prototype and now present experimental results that demonstrate its effectiveness.

Our main contributions are summarized as follows:

- A flexible multidimensional framework for evaluating recommender systems.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

- A comprehensive procedure for efficient development of the framework in order to support analysis of both, dataset facets and algorithms' performance using interactive OLAP queries (e.g., drill-down, slice, dice).
- The consideration of an extended set of evaluation measures, compared to standards such as the RMSE.
- Experimental results with intuitive outcomes based on swift visual analysis.

2. RELATED WORK

For general analysis of recommender systems, Breese [5] and Herlocker et al. [11] provide a comprehensive overview of evaluation measures with the aim of establishing comparability between recommender algorithms. Nowadays, the generally employed measures within the prevailing recommender tasks are MAE, (R)MSE, precision, recall, and F-measure. In addition further measures including confidence, coverage and diversity related measures are discussed but not yet broadly used. Especially the latter two have attracted attention over the last years as it is still not certain whether today's predictive accuracy or precision and recall related measures correlate directly with interestingness for a system's end users. As such various authors proposed and argued for new evaluation measures [22, 21, 6]. Ziegler [22] has analyzed the effect of diversity with respect to user satisfaction and introduced topic diversification and intra-list similarity as concepts for the recommender system community. Zhang and Hurley [21] have improved the intra-list similarity and suggested several solution strategies to the diversity problem. Celma and Herrera [6] have addressed the closely related novelty problem and propose several technical measures for coverage and similarity of item recommendation lists. All these important contributions focus on reporting single aggregate numbers per dataset and algorithm. While our framework can deliver those, too, it goes beyond that by its capability of combining the available measures and, most importantly, dissecting them among one or more dimensions.

Analysis of the end users' response to recommendations and their responses' correlation with the error measures used in research belongs to the field of Human-Recommender Interaction. It is best explored by user studies and large scale experiments, but both are very expensive to obtain and thus rarely conducted and rather small in scale. Select studies are [13, 14, 4]. Though in the context of classical information retrieval, Joachims et al [13] have conducted a highly relevant study on the biasing effect of the position an item has within a ranked list. In the context of implicit feedback vs. explicit feedback Jones et al [14] have conducted an important experiment on the preferences of users concerning recommendations generated by unobtrusively collected implicit feedback compared to recommendations based on explicitly stated preferences. Bollen et al. [4] have researched the effect of recommendation list length in combination with recommendation quality on perceived choice satisfaction. They found that for high quality recommendations, longer lists tend to overburden the user with difficult choice decisions. Against the background of those results we believe that for initial research on a dataset, forming an idea, checking if certain effects are present, working on collected data with a framework like the one presented is an acceptable proxy. With findings gained in this process, conducting meaningful user studies is an obvious next step.

Recent interesting findings with respect to dataset characteristics are e.g. the results obtained during the Netflix challenge [3, 17] on user and item base- effects and time-effects in data. When modeled appropriately, they have a noteworthy effect on recommender performance. The long time it took to observe these properties of the dataset might be an indicator for the fact that with currently available tools proper analysis of the data at hand is more difficult and tedious than it should be. This motivates the creation of easy-to-use tools enabling thorough analysis of the datasets and the recommender algorithm's results and presenting results in an easy to consume way for the respective analysts.

Notable work regarding the integration of OLAP and recommender systems stems from the research of Adomavicius et al. [2, 1]. They treat the recommender problem setting with its common dimensions of users, items, and rating as inherently multidimensional. But unlike this work, they focus on the multidimensionality of the generation of recommendations and on the recommenders themselves being multidimensional entities that can be queried like OLAP cubes (with a specifically derived query language, RQL). In contrast, our work acknowledges the multidimensional nature of recommender systems, but focusses on their multidimensional evaluation.

Existing frameworks for recommender systems analysis usually focus on the automatic selection of one recommendation technique over another. E.g., [10] is focussed on an API that allows retrieval and derivation of user satisfaction with respect to the recommenders employed. The AWESOME system by Thor and Rahm [20], the closest approach to that presented here, shares the data warehousing approach, the description of the necessary data preparation (ETL), and the insight of breaking down the measures used for recommender performance analysis by appropriate categories. But contrary to the approach presented here, the AWESOME framework is solely focussed on website performance and relies on static SQL-generated reports and decision criteria. Furthermore, it incorporates no multidimensional approach and does not aim at simplifying end-user-centric analysis or interactive analysis at all.

3. FRAMEWORK REQUIREMENTS

3.1 The Role of a Multidimensional Model

Business analysts expect all data of a recommender systems (information about items, generated recommendations, user preferences, etc.) to be organized around business entities in form of dimensions and measures based on a multidimensional model. A multidimensional model enforces structure upon data and expresses relationships between data elements [19]. Such a model, thus, allows business analysts to investigate all aspects of their recommender system by using the popular OLAP technology [7]. This technology provides powerful analytical capabilities that business analysts can query to detect trends, patterns and anomalies within the modeled measures of recommender systems' performance across all involved dimensions.

Multidimensional modeling provides comprehensibility for the business analysts by organizing entities and attributes of their recommender systems in a parent-child relationship (1:N in databases terminology), into dimensions that are

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

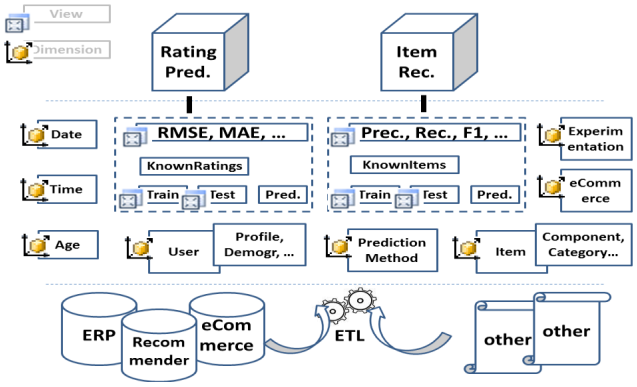

Figure 1: The recommender evaluation framework. The dimensions specified are connected with both fact table groups (dashed boxes in the center) and are thus available in both resulting cubes. End users can connect to the Rating Prediction and Item Recommendation cubes.

identified by a set of attributes. For instance, the dimension of recommended items may have as attributes the name of the product, its type, its brand and category, etc. For the business analyst, the attributes of a dimension represent a specific business view on the facts (or key performance indicators), which are derived from the intersection entities. The attributes of a dimension can be organized in a hierarchical way. For the example of a dimension about the user of the recommender systems, such a hierarchy can result from the geographic location of the user (e.g., address, city, or country). In a multidimensional model, the measures (sometimes called facts) are based in the center with the dimensions surrounding them, which forms the so called star schema that can be easily recognized by the business analysts. The star schema of the proposed framework will be analyzed in the following section.

It is important to notice that aggregated scores, such as the RMSE, are naturally supported. Nevertheless, the power of a multidimensional model resides in adding further derived measures and the capability of breaking all measures down along the dimensions defined in a very intuitive and highly automated way.

3.2 Core Features

Organizing recommender data in a principled way provides automation and tool support. The presented framework enables analysis of all common recommender datasets. It supports both rating prediction and item recommendation scenarios. Besides that, data from other application sources can and should be integrated for enriched analysis capabilities. Notable sources are ERP systems, eCommerce systems and experimentation platform systems employing recommender systems. Their integration leverages analysis of the recommender data by the information available within the application (e.g., recommender performance given the respective website layouts) and also analysis of the application data by recommender information (e.g., revenue by recommender algorithm).

Compared to RMSE, MAE, precision, recall, and F-measure, more information can be obtained with this framework as, first, additional measures e.g. for coverage, novelty, diversity analysis are easily integrated and thus available for all datasets. Second, all measures are enhanced by the respective ranks, (running) differences, (running) percentages, totals, standard deviations and more.

While a single numerical score assigned to each recommender algorithm's predictions is crucial for determining winners in challenges or when choosing which algorithm to deploy [8], from an business insight point of view a lot of interesting information is forgone this way. Relationships between aspects of the data and their influence on the measure may be hidden. One such may be deteriorating increase in algorithmic performance with respect to an increasing number of rating available per item, another the development of the average rating over the lifetime of an item in the product catalog. A key capability of this framework is exposing intuitive ways for analyzing the above measures by other measures or related dimensions.

From a usability point of view, this framework contributes convenient visual analysis empowering drag-drop analysis and interactive behavior. Furthermore, convenient visual presentation of the obtained results is integrated from the start as any standard conforming client can handle it. Manual querying is still possible as is extending the capabilities of the framework with custom measures, dimensions, or functions and post-processing of received results in other applications. Inspection of the original source data is possible via custom actions which allow the retrieval of the source rows that produced the respective result. Last but not least, aggregations allow for very fast analysis of very large datasets, compared to other tools.

The following section elaborates on the architecture of the multidimensional model that is used by the proposed framework, by providing its dimensions and measures.

4. THE ARCHITECTURE OF THE MULTIDIMENSIONAL FRAMEWORK

Figure 1 gives an overview of the architecture of the framework. The source data and the extract-transform-load (ETL) process cleaning it and moving it into the data store are located at the bottom of the framework. The middle tier stores the collected information in a data warehouse manner regarding facts (dashed boxes in the center) and dimensions (surrounding the facts). The multidimensional cubes (for rating recommendation and item prediction) sitting on top of the data store provide access to an extended set of measures (derived from the facts in the warehouse) that allow automatic navigation along their dimensions and interaction with other measures.

4.1 The Data Flow

The data gathered for analysis can be roughly divided into two categories:

***Core data*:** consisting of the algorithms' training data, such as past ratings, purchase transaction information, online click streams, audio listening data, ... and the persisted algorithms' predictions.

***Increase-insight data*:** can be used as a means to leverage the analytic power of the framework. It consists roughly of user master data, item master data, user transactional statistics, and item transactional statistics. This data basically captures the metadata and

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

usage statistics data not directly employed by current recommender algorithms (such as demographic data, geographic data, customer performance data...).

In case of recommender algorithms employed in production environments, relational databases housing the transactional system (maybe driving an e-commerce system like an ERP system or an online shop) will store rich business master data such as item and user demographic information, lifetime information and more, next to rating information, purchase information, and algorithm predictions. In case of scientific applications, different text files containing e.g. rating information, implicit feedback, and the respective user and item attributes for training and the algorithms' predictions are the traditional source of the data.

From the respective source, the master data, the transactional data, and the algorithm predictions are cleaned, transformed, and subsequently imported into a data warehouse. Referential integrity between the elements is maintained, so that e.g. ratings to items not existing in the system are impossible. Incongruent data is spotted during insert into the recommender warehouse and presented to the data expert.

Inside the framework, the data is logically split into two categories: measures (facts) that form the numeric information for analysis, and dimensions that form the axes of analysis for the related measures. In the framework schema (figure 1), the measures are stylized within the dashed boxes. The dimensions surrounding them and are connected to both, the rating prediction and the item recommendation measures.

4.2 The Measures

Both groups of measures analyzed by the framework—the measures for item recommendation algorithms and the measures for rating prediction algorithms—can be divided into basic statistical and information retrieval measures.

***Statistical measures*:** Among the basic statistical measures are counts and distinct counts, ranks, (running) differences and (running) percentages of various totals for each dimension table, train ratings, test ratings and predicted ratings; furthermore, averages and their standard deviations for the lifetime analysis, train ratings, test ratings, and predicted ratings.

***Information retrieval measures*:** Among the information retrieval measures are the popular MAE and (R)MSE for rating prediction, plus user-wise and item-wise aggregated precision, recall and F-measure for item prediction. Novelty, diversity, and coverage measures are also included as they provide additional insight. Furthermore, for comparative analysis, the differences in the measures between any two chosen (groups of) prediction methods are supported as additional measures.

In case a recommender system and thus this framework is accompanied by a commercial or scientific application, this application usually will have measures of its own. These measures can easily be integrated into the analysis. An example may be an eCommerce application adding sales measures such as gross revenue to the framework. These external measures can interact with the measures and the dimension of the framework.[1]

[1]E.g., the revenue could be split up by year and recommendation method, showing the business impact of a recommender.

4.3 The Dimensions

The dimensions are used for slicing and dicing the selected measures and for drilling down from global aggregates to fine granular values. For our framework, the dimensions depicted in figure 1 are:

***Date*:** The Date dimension is one of the core dimensions for temporal analysis. It consists of standard members such as Year, Quarter, Month, Week, Day and the respective hierarchies made up from those members. Furthermore, Year-to-date (YTD) and Quarter/Month/Week/Day of Year logic provides options such as searching for a Christmas or Academy Awards related effect.

***Time*:** The Time dimension offers Hour of Day and Minute of Day/Hour analysis. For international datasets this dimension profits from data being normalized to the time zone of the creator (meaning the user giving the rating).

***Age*:** The Age dimension is used for item and user lifetime analysis. Age refers to the relative age of the user or item at the time the rating is given/received or an item from a recommendation list is put into a shopping basket and allows for analysis of trends in relative time (c.f. section 6).

***User*:** User and the related dimensions such as UserProfile and UserDemographics allow for analysis by user master data and by using dynamically derived information such as activity related attributes. This enables grouping of the users and content generated by them (purchase histories, ratings) by information such as # of ratings or purchases, # of days of activity, gender, geography...

***Item*:** Item and the related dimensions such as ItemCategory and ItemComponent parallel the user-dimensions. In a movie dataset, the item components could be, e.g., actors, directors, and other credits.

***Prediction Method*:** The Prediction Method dimension allows the OLAP user to investigate the effects of the various classes and types or recommender systems and their respective parameters. Hierarchies, such as Recommender Class, Recommender Type, Recommender Parameters, simplify the navigation of the data.

***eCommerce*:** As recommender algorithms usually accompany a commercial or scientific application (e.g., eCommerce) having dimensions of its own, these dimensions can easily be integrated into and be used by our framework.

***Experimentation*:** In case this framework is used in an experiment-driven scenario [8], such as an online or marketing setting, Experimentation related dimensions should be used. They parallel the PredictionMethod dimension, but are more specific to their usage scenario.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

5. PROTOTYPE DESCRIPTION

This section describes the implementation of a research prototype for the proposed framework. The prototype was implemented using Microsoft SQL Server 2008 [18] and was used later for our performance evaluation.

In our evaluation, the prototype considers the Movielens 1m dataset [9], which is a common benchmark for recommender systems. It consists of 6.040 users, 3.883 items, and 1.000.209 ratings received over roughly three years. Each user has at least 20 ratings and the metadata supplied for the users is userId, gender, age bucket, occupation, and zipcode. Metadata for the item is movieId, title and genre information.

Following a classical data warehouse approach [15, 12], the database tables are divided into dimension and fact tables. The dimension tables generally consist of two kinds of information: static master data and dynamic metadata. The static master data usually originates from an ERP system or another authoritative source and contains e.g. naming information. The dynamic metadata is derived information interesting for evaluation purposes, such as numbers of ratings given or time spent on the system. To allow for an always up to date and rich information at the same time, we follow the approach of using base tables for dimension master data and views for dynamic metadata derived through various calculations. Further views then expose the combined information as pseudo table. The tables used in the warehouse of the prototype are Date, Time, Genre (instantiation of Category), Item, ItemGenre (table needed for mapping items and genres), Numbers (a helper table), Occupation, PredictedRatings, PredictedItems, PredictionMethod, TestRatings, TestItems, TrainRatings, TrainItems, and User. The Item and User table are in fact views over the master data provided with the Movielens dataset and dynamic information gathered from usage data. Further views are SquareError, UserwiseFMeasure, AllRatings, and AgeAnalysis.

On top of the warehouse prototype, an OLAP cube for rating prediction was created using Microsoft SQL Server Analysis Services. Within this cube, the respective measures were created: counts and sums, and further derived measures such as distinct counts, averages, standard deviations, ranks, (running) differences and (running) percentage. The core measures RMSE and MAE are derived from the error between predicted and actual ratings. The most important OLAP task with respect to framework development is to define the relationships between the measures and dimensions, as several dimensions are linked multiple times (e.g. the Age dimension is role-playing as it is linked against both item age and user age) or only indirect relationships exist (such as between category and rating the relationship is only established via item). Designing the relationships has to be exercised very carefully, as both correctness of the model and the ability to programmatically navigate dimensions and measures (adding them on the report axes, measure field or as filters) depend on this step. Linking members enables generic dimensions such as Prediction Method A, and Prediction Method B, that can be linked to chosen dimension members. This renders unnecessary the creation of the $n(n-1)/2$ possible measures yielding differences between any two prediction methods A and B (for, say, RMSE or F-measure). Furthermore, this approach allows choosing more than one dimension member, e.g. several runs of one algorithm with different parameters, as one linked member for aggregate analysis.

Before we go on to the evaluation of our prototype, let us state that our framework describes more than simply a model for designing evaluation frameworks. The prototype serves well as a template for other recommender datasets, too. With nothing changed besides the data load procedure, it can be used directly for, e.g., the other Movielens datasets, the Netflix challenge dataset or the Eachmovie dataset. Additional data available in those datasets (e.g. the tagging information from the Movielens 10m dataset) are either ignored or require an extension of the data warehouse and the multidimensional model (resulting in new analysis possibilities).

6. PERFORMANCE EVALUATION

In the previous section we have described the implementation of a research prototype of the proposed framework using the Movielens 1m dataset. Building on this prototype, we proceed with presenting a set of results that are obtained by applying it.

We have to clarify that the objective of our experimental evaluation is not limited to the comparison of specific recommender algorithms, as it is mostly performed in works that propose such algorithms. Our focus is, instead, on demonstrating the flexibility and easiness with which we can answer important questions for the performance of recommendations. It is generally agreed that explicitly modelling the effects describing changes in the rating behavior over the various users (user base-effect), items (item base-effect), and age of the respective item or user (time effects) [3, 16, 17]. For this reason, we choose to demonstrate the benefits of the proposed framework by setting our scope on those effects followed by exemplary dissecting the performance of two widely examined classes of recommender algorithms, i.e., collaborative filtering and matrix factorization. We also consider important the exploratory analysis of items and users, which can provide valuable insights for business analysts about factors determining the performance of their recommender systems. We believe that the results presented in the following demonstrate how easy it is to obtain them by using the proposed framework, which favors its usage in real-world applications, but also can provide valuable conclusions to motivate the usage of the framework for pure research purpose, since it allows for observing and analyzing the performance by combining all related dimensions that are being modeled.

All results presented in the remainder of this section could easily be obtained graphically by navigating the presented measures and dimensions using Excel 2007 as multidimensional client.

6.1 Exploratory Data Analysis

Using the framework, the first step for a research and a business analytics approach is exploring the data. As an example, the Calendar dimension (Date) is used to slice the average rating measure. Figure 2 presents this as pivot chart. The sharp slumps noticeable in March and August 2002 together with a general lack of smoothness beyond mid 2001 arouse curiosity and suggest replacing average rating by rating count (figure not shown). Changing from counts to running percentages proves that about 50 percent of the ratings in this dataset are spent within the first six months

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

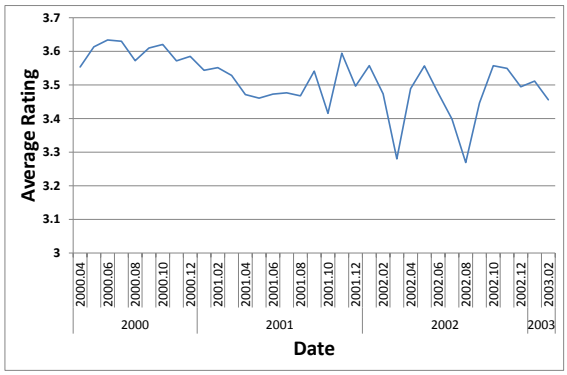

Figure 2: Average rating by date

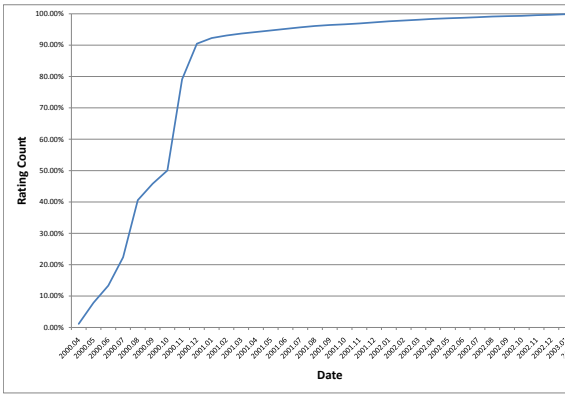

Figure 3: Rating count by date (running percentages)

out of nearly three years. Within two more months 90 percent of the ratings are assigned, roughly seven percent of the data for 50 percent of the time (figure 3).

6.1.1 *Item Analysis*

The framework allows an easy visualization of the item effect described e.g. in [16], namely that there usually is a systematic variation of the average rating per item. Aditionally, other factors can easily be integrated in such an analysis. Figure 4 shows the number of ratings received per item sorted by decreasing average rating. This underlines the need for regularization when using averages, as the movies rated highest only received a vanishing number of ratings.

Moving on the x-axis from single items to rating count buckets containing a roughly equal number of items, a trend of heavier rated items being rated higher can be observed (figure omitted for space reasons). A possible explanation

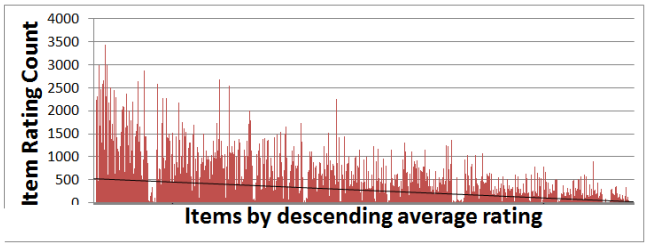

Figure 4: Item rating count sorted by decreasing average rating

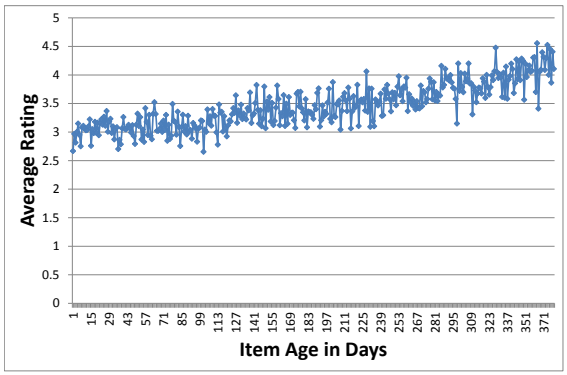

Figure 5: The all-time classics effect. Ratings tend to increase with the age of the movie at the time the rating is received. Age is measured in time since the first rating recorded.

might be that blockbuster movies accumulate a huge number of generally positive ratings during a short time and the all-time classics earn a slow but steady share of additional coverage. That all-time classics receive higher ratings can nicely be proved with the framework, too. Consistent with findings during the final phase of the Netflix competition by Koren [17], figure 5 shows a justification for the good results obtained by adding time-variant base effects to recommender algorithms. Besides the all-time classics effect, the blockbuster effect can also be observed (figure 6), showing that items who receive numerous ratings per day on average also have a higher rating.

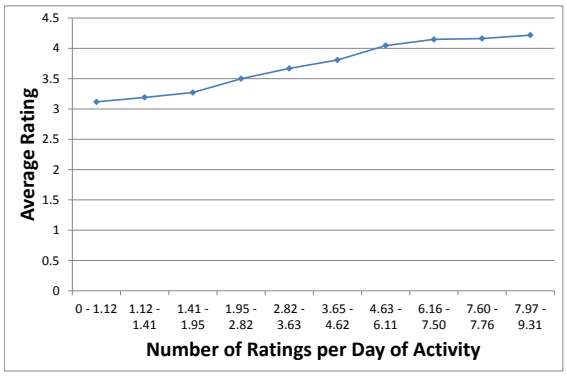

Figure 6: The blockbuster effect. Increasing average item rating with increasing number of ratings received per day.

Slicing the average rating by Genre shows a variation among the different genre with Film-Noir being rated best (average rating 4.07, 1.83% of ratings received), and Horror being rated worst (3.21, 7,64%). Of the Genres with at least ten percent of the ratings received Drama scores highest (3.76, 34.45%) and Sci-Fi lowest (3.46, 15.73%). Figure not shown.

6.1.2 *User Analysis*

The user effect can be analyzed just as easy as the item effect. Reproducing the analysis explained above on the users,

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

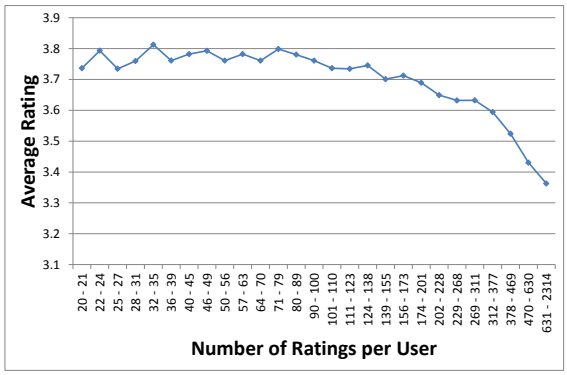

Figure 7: The effect of the number of ratings per user on the average rating

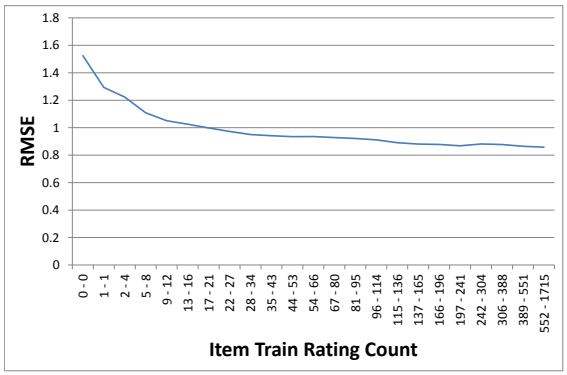

Figure 8: Item rating count effect on a factor model. Buckets created on roughly equal item count.

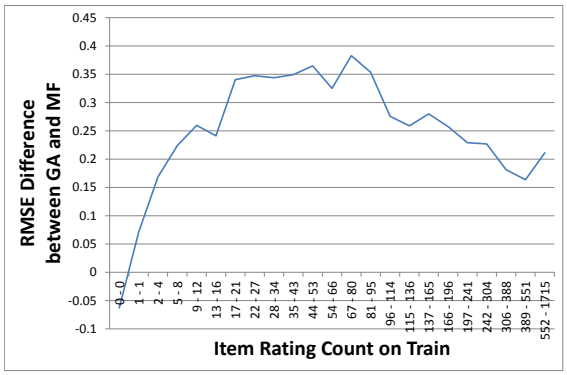

Figure 9: Difference in RMSE between Matrix Factorization (MF) and Global Average (GA) vs. ratings available per item on the train dataset.

it is interesting to notice that for heavy raters the user rating count effect is inverse to the item rating count effect described above (figure 7): the higher the amount of ratings spent by a given user, the lower his or her average rating. One explanation to this behavior might be that real heavy raters encounter a lot of rather trashy or at least low quality movies.

6.2 Recommender Model Diagnostics

For algorithm performance comparison, the Movielens 1m ratings were randomly split into two nearly equal size partitions, one for training (500103), and one for testing (500104 ratings). Algorithm parameter estimation was conducted on the training samples only, predictions were conducted solely on the test partition. Exemplarily, a vanilla matrix factorization (20 features, regularization 0.09, learn rate 0.01, 56 iterations, hyperparameters optimized by 5-fold cross-validation) is analyzed.[2]

For a researcher the general aim will be to improve the overall RMSE or F-Measure, depending on the task, as this is usually what wins a challenge or raises the bar on a given dataset. For a business analyst this is not necessarily the case. A business user might be interested in breaking down the algorithm's RMSE over categories or top items or top users as this may be relevant information from a monetary aspect. The results of the respective queries may well lead to one algorithm being replaced by another on a certain part of the dataset (e.g. subset of the product hierarchy).

In figure 8, RMSE is plotted vs. item rating count in train.This indicates that more ratings on an item do help factor models. Interpreted the other way around, for a business user, this implies that this matrix factorization yields best performance on the items most crucial to him from a top sales point of view (though for slow seller other algorithms might be more helpful).

The same trend can be spotted when RMSE is analyzed by user rating count on the training set (figure omitted for space reasons), though the shape of the curve follows a straighter line than for the item train rating count (where it follows more an exponential decay).

Due to the approach taken in the design of the OLAP cube the number of recommender algorithms comparable as A and B is not limited; neither does it have to be exactly one algorithm being compared with exactly one other, as multiple selection is possible. Furthermore—given the predictions are already in the warehouse—replacing one method by another or grouping several methods as A or B can nicely be achieved by selecting them in the appropriate drop-down list. Exemplarily, the matrix factorization analyzed above is compared to the global average of ratings as baseline recommendation method. Figure 9 reveals that for this factor model more ratings on train do increase the relative performance, as expected, up to a point from which the static baseline method will gain back roughly half the lost ground. Investigation of this issue might be interesting for future recommender models.

All results presented could be obtained very fast: when judging the time needed to design query and report (chart)—which was on average seconds for construction of the query and making the chart look nice—, and when judging execution time—which was in the sub-second timeframe.

[2]The matrix factorization yielded an RMSE of 0.8831 given the presented train-test split.

7. CONCLUSIONS

We have proposed a novel multidimensional framework for integrating OLAP with the challenging task of evaluating recommender systems. We have presented the architecture of the framework as a template and described the implementation of a research prototype. Consistent with the other papers at this workshop, the authors of this work

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper6.pdf

believe that the perceived value of a system largely depends on its user interface. Thus, this work provides an easy to use framework supporting visual analysis. Our evaluation demonstrates, too, some of the elegance of obtaining observations with the proposed framework. Besides showing the validity of findings during the recent Netflix prize on another dataset, we could provide new insights, too. With respect to the recommender performance evaluation and the validity of RMSE as an evaluation metric, it would be interesting to see if a significant difference in RMSE concerning the amount of ratings present in the training set would also lead to significant effects in a related user study.

In our future work, we will consider the extension of our research prototype and develop a web-based implementation that will promote its usage.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge the co-funding of their work through the European Commission FP7 project MyMedia (grant agreement no. 215006) and through the European Regional Development Fund project LEFOS (grant agreement no. 80028934).

9. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: A data warehousing approach. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 180–192, London, UK, 2001. Springer-Verlag.

[3] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM.

[4] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *RecSys '10: Proceedings of the 2010 ACM conference on Recommender systems*, New York, NY, USA, 2010. ACM.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *MSR-TR-98-12*, pages 43–52. Morgan Kaufmann, 1998.

[6] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186, New York, NY, USA, 2008. ACM.

[7] E. Codd, S. Codd, and C. Salley. Providing OLAP to user-analysts: An it mandate. Ann Arbor,MI, 1993.

[8] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, New York, NY, USA, 2009. ACM.

[9] GroupLens. Movielens data sets. http://www.grouplens.org/node/73.

[10] C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*. Springer Verlag, 2002.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[12] W. H. Inmon. *Building the Data Warehouse*. Wiley, 4th ed., 2005.

[13] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[14] N. Jones, P. Pu, and L. Chen. How users perceive and appraise personalized recommendations. In *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 461–466, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, 2nd ed., 2002.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

[18] Microsoft. Microsoft SQL Server 2008 homepage. http://www.microsoft.com/sqlserver/2008/.

[19] J. O'Brien and G. Marakas. *Management Information Systems*. McGraw-Hill/Irwin, 9th ed., 2009.

[20] A. Thor and E. Rahm. Awesome: a data warehouse-based system for adaptive website recommendations. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 384–395. VLDB Endowment, 2004.

[21] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, New York, NY, USA, 2008. ACM Press.

[22] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

Recommendations for End-User Development

Will Haines
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
1 650-859-6153
haines@ai.sri.com

Melinda Gervasio
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
1 650-859-4411
gervasio@ai.sri.com

Aaron Spaulding
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
1 650-859-3911
spaulding@ai.sri.com

Bart Peintner
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
1 650-859-3209
peintner@ai.sri.com

ABSTRACT

End-user development (EUD), the practice of users creating, modifying, or extending programs for personal use, is a valuable but often challenging task for nonprogrammers. From the beginning, EUD systems have shown that recommendations can improve the user experience. However, these usability improvements are limited by a reliance on handcrafted rules and heuristics to generate reasonable and useful suggestions. When the number of possible recommendations is large or the available context is too limited for traditional reasoning techniques, recommender technologies present a promising solution. In this paper, we provide an overview of the state of the art in end-user development, focusing on the different kinds of recommendations made to users. We identify four classes of suggestion that could most directly benefit from existing recommendation techniques. Along the way we explore straightforward applications of recommender algorithms as well as a few difficult but high-value recommendation problems in EUD. We discuss the ways that EUD systems have been evaluated in the past and suggest the modifications necessary to evaluate recommenders within the EUD context. We highlight EUD research as one area that can facilitate the transition of recommender system evaluation from algorithmic performance evaluation to a more user-centered approach. We conclude by restating our findings as a new set of research challenges for the recommender systems community.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *end-user development, recommender systems.*

General Terms

Algorithms, Design, Experimentation, Human Factors.

Keywords

end-user development, recommender systems.

1. INTRODUCTION: THE CASE FOR END-USER DEVELOPMENT

Computing devices are ubiquitous in today's professional environments and are increasingly invading our homes and mobile lives. Unfortunately, a deep understanding of these systems, and the ability to modify them, remains confined to the realm of the specialist. While the human-computer interaction community has made great strides in improving software usability, it has devoted far less attention to making systems customizable by end-users [22]. We argue here that existing recommender techniques can make a meaningful contribution toward increasing the usability, and therefore the acceptance and proliferation, of customizable software.

The current state of the art in user-centered software design is to engage users in an iterative design process and to test systems with users to refine the interaction design. Customization, if available, is built into the system at design time as a bounded number of user-selectable options. However, as workflows and processes change over time, even customized software applications need updates. Currently, the burden of these updates falls on the shoulders of professional developers, but the pool of users needing customizations is expected to grow much faster than the supply of professional software engineers [3].

One promising approach is end-user development (EUD), the practice of users creating, modifying, or extending programs for personal use [22,18]. This approach has two main benefits. One, it puts systems design in the hands of the domain experts who are most familiar with what needs should be met. Two, it scales with both a rapid increase in users and the increasing rate of change of many business processes. Unfortunately, EUD faces one major challenge—most end users do not have the specialized knowledge currently required to perform even basic development tasks [25].

As such, EUD research mainly focuses on approaches for lowering the barrier of entry to software development. Such approaches cover a wide spectrum, from enhancing the macros and spreadsheets that millions use every day to sophisticated algorithms that create programs by example without ever exposing the user to textual code [22]. While the technology behind these approaches may vary a great deal, there are some crosscutting techniques that seem to improve usability across the spectrum of EUD systems. In this paper, we will discuss one particular mechanism for improving user performance—system-generated recommendations.

As early as 1991, EUD systems like EAGER were using simple proactive suggestions as a component of their user interaction [6]. When the system detects that the user is performing an iterative task, it suggests a sequence of actions for completing the iteration automatically. In this case, the recommendation algorithm is straightforward, owing to the fact that there is only one recommendation type. If the system can complete the iteration it makes the recommendation, otherwise it does not.

Nearly twenty years later, advances in EUD systems have greatly expanded the opportunities for offering recommendations; however, most approaches continue to rely on constrained forms of recommendation that use handcrafted rules to make limited types of suggestions. In this paper, we explore the different classes of recommendations made in EUD systems, highlighting the areas where current approaches fall short and how recommender technologies can help fill the gap. Concurrently, we emphasize the ways in which these systems have been evaluated to date and discuss the ways in which such approaches will need

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

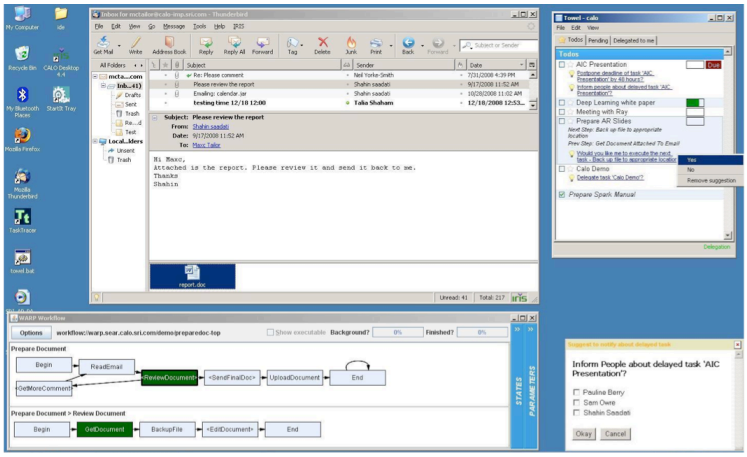

Figure 1. WARP identifies that the user is delayed in creating a document and offers to help [33].

modification to successfully evaluate integrated recommenders. We believe that by evaluating recommender systems in a user-focused context like EUD, researchers can facilitate the transition of recommender system evaluation from a focus on algorithm performance evaluation to a more user-centered approach.

2. INTEGRATING AUTOMATION INTO THE USER'S WORKFLOW

A core tenet of user-centered design is to create systems that mesh with users' existing workflows and environments [3]. EAGER is an early example of a tool that incorporates system suggestions to help "fit [end-user development] into the user's existing workflows" [18]. The focus on workflow integration is very important; it is likely that users will ignore an EUD system whose barrier to entry is too high. One approach to workflow integration, suggested by Lieberman, is to make "the cognitive load of switching from using to adapting … as low as possible" [22]. One way EUD systems have achieved this low barrier is by offering to automate portions of the user's workflow, essentially bypassing an explicit programming process and attempting to directly accomplish the user's task instead.

2.1 Current Approaches

One of the earliest approaches to EUD was programming by demonstration (PBD), also known as programming by example [5,21]. Many PBD systems rely on users explicitly demonstrating the process to be automated. However, some systems instead rely on implicit examples, continuously observing the user's actions to find repetitions over which they can learn a looping program to complete the user's task. Examples include EAGER, Dynamic Macro, and APE [6,24,28]. By recommending automation directly within the user's workflow, these systems achieve EUD transparently, without the user's awareness of having programmed the system. However, they are limited to automating repetitive tasks within the space of the looping programs they can generate.

A more general approach to automation within the user's workflow relies on activity recognition to observe what the user is doing and infer what that user is trying to accomplish. In combination with some mechanism for determining appropriate assistance, the system can use activity recognition to assist with the completion of a task. For example, Lumière uses Bayesian user models to offer context-dependent assistance [13]. While Lumière can offer assistance on a wider variety of tasks than the PBD systems focused on repetitive tasks, it is limited to assisting the tasks encoded by the developers, unlike PBD systems, which acquire looping programs on the fly.

Some systems combine aspects of both approaches. WARP (Figure 1), like Lumière, utilizes probabilistic models for activity recognition on a wide variety of tasks [33]. However, its meta-level assistance patterns are designed to work over a knowledge base of procedures rather than a developer-defined set of tasks. Because of this, the system not only can offer to automate more complex tasks but, through EUD, can continue to extend its knowledge base to handle a wider variety of tasks.

Task Assistant is another system that makes recommendations over an extensible knowledge base [27]. It allows users to explicitly define a workflow that groups of users collaboratively execute. It promotes automation by allowing the user to manually attach automated procedures, often produced by EUD, that support individual tasks in the workflow. Task Assistant uses these manual attachments to inform its suggestions for attaching other automated procedures to future tasks.

2.2 Recommender Systems Opportunities

EUD systems that make automation recommendations within a user's workflow provide a gentle transition from the user's core workflow into the world of programming, as users are generally not even aware that programs are being created or selected for execution. While current systems already provide useful EUD assistance, the use of recommender technology raises the possibility of further improvement.

2.2.1 Recommending shared procedures

EUD systems such as WARP and Task Assistant, which utilize a potentially unbounded set of automated procedures, offer the greatest opportunity for the application of recommender technology. Consider a shared procedure repository for the members of an organization. As the library of assistive procedures grows, handcrafting the rules or patterns for recommendation becomes more difficult. The problem is exacerbated when there are multiple criteria for evaluating procedure quality or applicability within a given context. As new procedure sources (e.g., web services, new EUD systems) continue to proliferate, the problem of finding and automating procedures and associating them to user tasks and workflows will only increase in frequency and importance.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

For example, suppose the system determines that a user wants to schedule a meeting with some coworkers. The procedure repository may contain dozens of meeting scheduling procedures—some idiosyncratic to a given organization, some specific to certain types of meetings, and some buggy or outdated. Even if the procedure repository was constrained only to meeting scheduling, it would be difficult to craft a set of rules that covered all situations. Using recommender technology, one could imagine leveraging information about what other users have found useful (or not) to make better decisions about what procedures to recommend. The challenge is in incorporating sufficient context from the user's current activity into the recommendation algorithms.

2.2.2 Improving activity recognition

Instead of applying recommender technology only after determining what the user is doing, one could also imagine a system like WARP applying a recommender within the activity recognition algorithm itself. In this case, the algorithm might be able to narrow its search space to the activities that similar people have automated or been assisted with in the past. Such an enhancement would be particularly valuable in the case where the number of identifiable activities is very high.

2.3 Evaluation

To evaluate a recommender's ability to improve an EUD system's level of integration into a user's workflow, system designers must take a more user-centered approach than is traditional in recommender system evaluation. Instead of focusing on algorithmic performance of independent predictions involving a single, primitive task, EUD evaluation must situate itself in the context of a user performing a task, often comprising multiple subtasks. For Dynamic Macro and APE, this meant considering task performance time and user acceptance as part of the success criteria for the application [24,28]. Other systems may be able perform post-hoc analysis on user logs and avoid addressing the user's workflow directly, but regardless of methodology, the type and complexity of the user's task to automate will have some effect on the evaluation results [9].

The tasks supported by EUD systems are generally more complex than the simple viewing or purchasing decisions assisted by a traditional recommender system, so task-oriented evaluation can become particularly tricky. When testing even a simple task, it is can be difficult to tease apart user interface and algorithmic concerns. When the task becomes more complex, a simple design mistake can limit the user's ability to complete a task, rendering analysis of a recommender's efficacy difficult.

In such cases, it is desirable to control for user interface variation using a two-step process. First, a series of short qualitative studies can quickly identify high-priority user interface problems that can confound later study results. We find that think-aloud and heuristic evaluation protocols are well suited for this purpose [31,26]. After resolving the issues identified qualitatively, one can perform a controlled experiment that compares the user interface with a naïve recommendation implementation against the same user interface backed by more sophisticated algorithms.

A final consideration for evaluating increased automation and activity recognition is that recommendations may occur infrequently in comparison to the entire duration of the user's workflow. In this situation, longitudinal study protocols are appropriate. In the PBD space, Dynamic Macros and LAPDOG both worked with logs collected over an extended period of time [24,8]. For activity recognition, such evaluations do not yet exist in the literature, but one possible approach is a hybrid diary/log study that captures both logging information about when a task is recognized appropriately and diary entries capturing instances where the user expected a recommendation and received none.

3. HELPING THE USER MAKE THE RIGHT DECISIONS

The cognitive burden on end-user developers can be further reduced, and their task performance improved, by providing recommendations that help them make better programming decisions. This approach can be of particularly high impact in EUD systems that require a mixed-initiative interaction with the user—with appropriate recommendations, the user and system can move through their dialog more quickly and with fewer errors. By using the recommendations to sort lists of programming decisions by likelihood of correctness, EUD systems can help users make the right decisions in an unobtrusive, easy-to-override fashion.

3.1 Current Approaches

The Integrated Task Learning (ITL) system provides a procedure editor that takes advantage of recommendations to produce sensible defaults [11]. For example, to edit the data flow in a procedure, a user clicks on the argument to edit, and the procedure editor provides a list of suggested changes (Figure 2). The suggestions in this case are based on reasoning about the procedure using scope and type information. This bounds the recommendation space to suggestions that would not result in an invalid procedure; however, the system currently makes no attempt to rank the suggestions in any other way.

Another task in EUD where sensible default behavior is important is the generalization performed by PBD systems [5,21]. Because users need to provide the demonstrations from which PBD systems learn, a primary challenge is to find the correct generalization with as few examples as possible. One way to do this is to involve the user in the generalization process by presenting candidate generalizations and letting the user pick the correct one. Past systems have explored several different methods for selecting the candidates to present to the user. For example, SMARTedit performs conservative generalization within a well-defined version space and then uses a probabilistic weighting scheme to rank alternative generalizations [38].

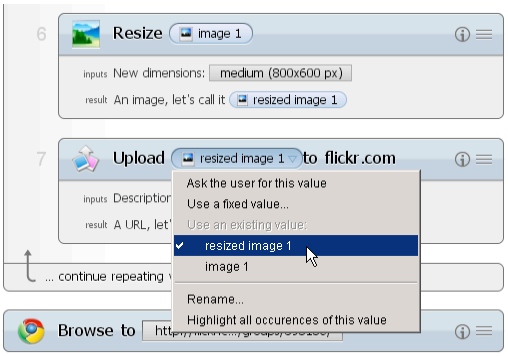

Figure 2. Clicking on 'resized image 1' in ITL provides suggestions under the heading 'Use an existing value:'.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

CHINLE uses a similar scheme not just to rank the candidates but also to color-code its interface [9]. LAPDOG uses heuristics to guide the inferencing it performs to explain connections between parameters and filter out unlikely generalizations [15].

3.2 Recommender Systems Opportunities

For this class of problem, recommender systems would likely sit behind the scenes, transparently organizing the plethora of options that systems currently present to the user. Potentially, this input could even cause systems to suppress options that are highly unlikely to be to be useful.

3.2.1 Suggesting preferred defaults

For a system like ITL, the primary limitation for reasoning is that for large procedures, the number of valid suggestions can be large, and if the user's desired edit does not appear high in the list of alternatives, that user's task performance can suffer. As with procedure recommendation, in large repositories, collaborative or social recommender techniques could also be used to improve recommended edits. However, a key difference is that the possible edits are information sparse compared to procedures. Algorithms would need to utilize the recommendation context even more to make appropriate suggestions, i.e. "other users *in this situation* tended to make change A" rather than "other users *generally like you* tended to make change A." Leveraging this context presents a challenge that recommender systems are only beginning to explore [1].

3.2.2 Suggesting more likely generalizations

The approach of presenting users with candidate generalizations and letting them choose the correct generalization is appealing because it transforms the difficult problem (for the system) of divining user intent into the relatively easier problem (for the user) of recognizing the correct generalization. However, with very few demonstrations, the number of alternative generalizations can become prohibitively large and determining the best ones may be difficult. PBD systems can thus benefit from mechanisms to organize the space of options and provide guidance toward reasonable generalizations.

Collaborative or social recommendations may help determine what and how to generalize. For example, in SMARTedit, a recommender system could speed up learning by suggesting a higher level of generalization based on how other similar actions were generalized in past procedures. And in LAPDOG, the preferences over different explanations could be derived from the explanations selected to explain similar actions in procedures developed by colleagues within an organization.

3.3 Evaluation

When determining the extent to which recommendations help users to make appropriate decisions, there are two phases in which a rigorous evaluation can be helpful: design time and implementation time. Post-implementation evaluations are common in the EUD literature but often fail to provide insight into the utility of the system in the field. Design evaluations are less common, although they are discussed for some systems, including ITL and CoScripter [11,20]. Such evaluations can define the types of recommendations that are beneficial to the user and to establish a set of ideal recommendations against which to evaluate actual algorithm performance in context.

For ITL, early application of think-aloud design evaluation methods proved useful to determine that recommended defaults would make a big difference in user performance [11]. Interestingly, it was possible to discover this insight using a wizard-of-oz protocol and thus the study required no actual recommender implementation. The lesson here is that it is possible to evaluate important EUD user interactions without having to settle on a recommender algorithm upfront.

In the space of recommending likely generalizations, evaluations have focused primarily on measuring whether the selected PBD algorithm can find the correct generalization—for example, in terms of whether it includes the correct generalization or if makes a correct predication using the generalization and ranks it highly [9,30,5]. In LAPDOG, there is an assumption that users will be able to recognize the correct generalization reasonably easily. However, many programming constructs, particularly when they include variables, are not easy for end users to comprehend. SMARTedit and CHINLE circumvent the generalization issue by presenting concrete predictions about the next step to execute in the context of a specific task. In this approach, users never have the sense of creating an actual program. Another interesting approach is *sloppy programming*, which represents programs in a pseudo-natural language that is interpretable by both humans and computers [22]. In general, PBD approaches that require the presentation of learned programs to users, whether for verification or for selection, need to identify the barriers to understanding such programs. For such cases, design-time survey methods may help to determine how users conceptualize the space of generalizations, leading system designers to more insight about how to create UI affordances to help users navigate the space of recommended generalizations.

4. HANDLING ERRORS

Much as is the case with professional programmers, end-user programmers spend a large proportion of their time debugging [29,14]. Consequently, many EUD systems provide mechanisms to help users identify and correct errors within their programs. Traditional debugging environments provide tools that allow developers to explore their code and track down errors; however, this exploration is left largely up to the user. A growing body of research suggests that for many users, this exploration process is hypothesis driven [10]. Unfortunately, when users do not have adequate knowledge of exactly how their programs execute, they can have difficultly formulating correct debugging hypotheses.

4.1 Current Approaches

Ko's Whyline provides a novel approach to supporting hypothesis-driven debugging by suggesting "Why?" questions for the user to explore when debugging (Figure 3) [17,15]. Its current implementation is more focused on professional developers than "end-user" developers, but it has been successfully implemented in novice programming environments and could be extended to pure-EUD systems [15].

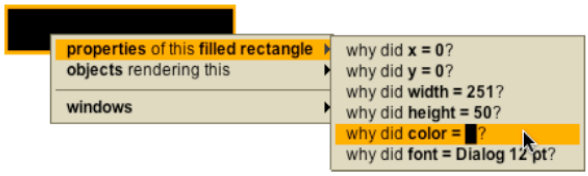

Figure 3. Whyline generates a set of possible questions about the black rectangle [17].

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

By recording and analyzing program execution and relating it back to the source code, Whyline is able to generate questions that focus on why a particular code segment behaved in a certain way rather than simply on why the program produced certain output [17]. By automatically identifying code related to failures, Whyline bounds the space of appropriate debugging hypotheses and helps programmers avoid guesses about irrelevant code segments.

Whyline generates its questions and answers automatically using heuristics and program slicing [16]. Questions are limited by two heuristics. The first heuristic relates to how users debug: hypotheses must reference observable failures. The second heuristic limits the number of possible entities to questions about code that the user wrote or directly referenced, with the expectation that the user will not ask questions about totally unfamiliar code. Once Whyline generates a set of questions, it extracts answers using program slicing to generate a "causal chain" of the code executed [16]. It then presents the questions to the user, fetching answers on demand.

Another approach to error handling is to take the act of debugging out of the user's hands entirely by automatically verifying the program for correctness before something goes wrong. For some classes of programming problems, EUD systems can detect errors and provide suggested fixes without forcing the user to search for problems. The ITL procedure editor takes this approach when possible (Figure 4) [11].

When the user performs an edit that causes an error, such as unbinding a variable, the editor detects the error via static analysis, marking the offending action. When the user clicks on the error icon, the system suggests solutions for that error. The solutions are parameterized to fit the given situation, but there are only a limited number of solutions at the present time [11].

4.2 Recommender Systems Opportunities

Whyline uses highly dynamic techniques to generate its hypothesis recommendations; however, it is still limited by its output and familiarity heuristics. Both exist to limit the number of questions and reduce time necessary to perform program slicing; however, as with all heuristics, they may not be correct all the time. By reducing the question space using recommendations rather than static heuristics, Whyline could find more bugs without sacrificing the processing time required to slice the program for every possible output primitive.

4.2.1 Suggesting potential problems based on similar programs

Users do not always test their programs on every possible execution path, so the observed output heuristic captures only a subset of the bugs that actually exist. A recommender that works over a community pool of debugged programs could suggest potential problem areas outside of the observed output based on problem areas in similar programs. This approach requires algorithms to be able to determine what features indicate programmatic similarity, a more difficult task than many traditional recommender applications. Nevertheless, programs are information-rich—the core problem is identifying what information is most relevant to determining similarity.

4.2.2 Suggesting examples

It is likely that users will be less successful debugging code with which they are unfamiliar. The familiarity heuristic weeds out

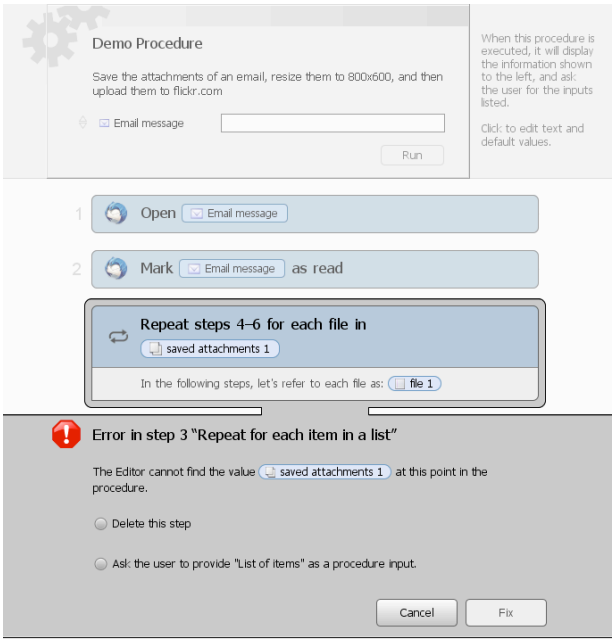

Figure 4. ITL alerts the user that 'saved attachments 1' is unbound and suggests two ways to fix the problem.

libraries that the user cannot modify without source code access, but it does not actually address the root cause of the unfamiliarity that leads to coding problems. A recommender of relevant code examples for problem application programming interfaces (APIs) could go a long way toward helping users debug calls into unfamiliar source. In fact, Brandt suggests that "opportunistic programming," based on web searches of code examples and tutorials, constitutes a primary programming approach for novice programmers [4]. His work on supporting this sort of programming focuses on how to improve user code browsing experiences, but recommendations do not make up the core of the approach. It may well be fruitful to see how system-driven recommendations interact with user-driven browsing to support opportunistic programmers.

4.2.3 Suggesting solutions to programming problems

Once errors are identified, recommending a subsequent fix is likely to be a highly relevant challenge for the recommender community. Here, current systems tend to pre-engineer relatively simple solutions to straightforward errors. If one could instead capture the ingenuity of an entire EUD system user community, it may be possible to propose solutions for more complex classes of problems. By creating a data set out of the problems and solutions faced by previous end-user developers, recommenders could suggest, and with some extension possibly apply, novel solutions to programming problems initially unanticipated by the creators of EUD systems. Such an advance would make EUD systems much more robust to unexpected programming challenges.

4.3 Evaluation

Whyline's evaluation consisted of a task-focused controlled experiment of debugging approaches [17]. Such a protocol also lends itself to the evaluation of error-handling recommenders because it provides a solid task context while still allowing the

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI),
Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

experimenter to isolate algorithmic performance from user-interface design choices. For Whyline, a traditional breakpoint debugging system served as the control condition, and user task completion defined the success criteria. Thus, the experiment tested hypotheses about how the interaction design paradigm would affect user task performance

To test the recommender opportunities above, one would instead control for user interface deviation and vary the recommendations while still measuring task completion to evaluate the success of each condition. To manage user interface variation, we again suggest following the two-step approach advocated in section 2.3, first isolating and solving confounding user interface problems qualitatively, and then comparing task performance on interface without recommendations against an interface with recommendations.

Other possible conditions to test include recommendation domains of varying size or composition. Likewise, testing tunable parameters like recommendation confidence or recommendation diversity could yield insights into exactly what sorts of error-handling recommendations provide the most value to end user developers.

5. SUPPORTING UNPLANNED SHARING

End-user programmers, by definition, do not set out to create programs for others; however, unplanned sharing is a frequent side effect of EUD [23]. Even if end users do not plan on sharing their programs, they can often benefit from using someone else's code. While many early EUD systems lacked a community and, by extension, lacked sharing, several systems now leverage this powerful capability.

5.1 Current Approaches

Two systems that benefit from sharing are ITL, which provides a demonstration-based EUD environment for collaborative military command and control, and Task Assistant, which allows users to explicitly encode best-practice workflows in a sharable way [7,27]. An even more widely used application is the CoScripter web automation system. Its relatively wide adoption provides a case study in how sharing can affect a community of end-user programmers [20]. While both ITL and CoScripter provide end users with a rich repository of programs to share, neither currently provides sharing support past simple browsing and search.

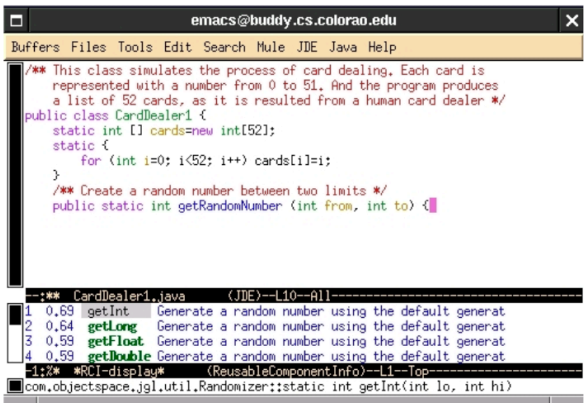

Figure 5. CodeBroker suggests that the user use the predefined getInt() method rather than the user's newly created getRandomNumber() method [32].

This sort of aid does not go far enough because users generally do not have sufficient familiarity with large codebases to know when code that solves their problems already exists [32]. Ye et al. suggest that "information push" (proactive recommendation) is one mechanism to assist users in such situations.

The CodeBroker system is one example of proactive development support. It recommends API calls to the user based on programming context (Figure 5) [32]. While it is not focused on "end-user" developers, it is straightforward to imagine using similar mechanisms for a code repository like the one used by CoScripter. CodeBroker generates its recommendations by gathering context from code comments. Then it applies Latent Semantic Analysis (LSA) to determine what components in the code base are conceptually similar to the concepts described by comments [19]. It further reduces the suggestion space by applying signature matching to only suggest API calls that are close type matches to the current calling context.

In the case that the above technique does not produce satisfactory results, the programmer can create a "discourse model" to manually identify suggestions that were not relevant [32]. Such irrelevant suggestions are not displayed in future queries. The system expands this concept to also include "user models" that manually identify code with which the user is already very familiar. Following the intuition that users do not want to get suggestions for familiar code, user models also serve to reduce the suggestion space for a given user. It is important to note, however, that in CodeBroker users manually create both the discourse and user models and must know a specific syntax for doing so.

5.2 Recommender Systems Opportunities

Community has not always been a central focus in EUD, but as recent systems push the boundaries toward more and more collaboration, community-based recommender systems seem to be a good fit. More users typically means more possible recommendations, and recommender algorithms can help EUD systems to cope with this increased scale.

5.2.1 Suggesting reusable code elements

LSA has been integrated previously into a collaborative filtering algorithm, so one could imagine implementing similar recommendation techniques within CodeBroker [12]. The system's major usability problem for end-user developers is likely to be its manually specified user and discourse models. Users are not apt to learn a brand-new programming syntax just to remove non-salient recommendations. If users can instead provide positive/negative feedback regarding the system recommendations, recommender technology could be used to acquire user profiles for making desirable recommendations based on individual user feedback as well as collective group feedback. In the latter case, by driving such a system with simple ratings rather than a complex programming syntax, users become more likely to actually provide the feedback necessary for the system to tailor its behavior to meet their preferences.

5.2.2 Assisting novices

Higher hurdles remain for EUD systems that are actually focused on novice programmers. Not only do these programmers not know what API calls exist, but they are also likely to misuse such calls. As discussed in Section 4.2.2, recommendations that also include code examples can be useful to teach novice opportunistic programmers how to reuse code.

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

5.2.3 Suggesting best practices

Suggesting code reuse is only half of the battle. End-user developers are unlikely to aim to create reusable code when their core goal is to program for personal use [18]. This is unfortunate, because by creating more reusable code, users can not only facilitate reuse by others but also make more robust, flexible programs for themselves [2]. It is difficult to get users to code to appropriate levels of abstraction, and current EUD systems have to attempt to build such abstractions into the development environment [18]. One possibility is to recommend code structures that reflect the code abstractions created by the most experienced programmers in an EUD community. Recommender technology could facilitate suggesting appropriate code abstractions to novice programmers working on programs similar to those already solved by experts.

The workflows in Task Assistant explicitly encode best practices, and therefore are ripe for sharing. The difficulty for recommender systems in this case is the possibly endless copies of very similar workflows. For example, in a large company or community, subgroups may have many similar, but distinct, workflows for hiring a new employee. The best workflow for a given user depends on factors such as the user's usage history, which other users have used particular versions, and the relationship between the current user and the other users. A hybrid of collaborative filtering and more personalized recommendation techniques promises to improve such recommendations.

5.3 Evaluation

Sharing is a particularly difficult user need to evaluate because a realistic evaluation must be situated within a community of users. CoScripter was able to overcome this difficultly by selecting users to interview based on indicators gathered from broad-based logging of user activities [20]. Unfortunately, this approach is difficult to replicate without a critical mass of users such as the one CoScripter enjoys. Without such a user pool, the CodeBroker evaluation protocol falls back on extrapolating code sharing efficacy from precision and recall of the method calls suggested [32]. While these measures are valuable, they cannot tell the whole story of how a community-based sharing system will operate in practice.

In a situation where a large user pool is not available, we suggest an approach that combines a longitudinal protocol with user satisfaction surveys at regular intervals. In this case, users have a longer time in which to benefit from sharing, and the evaluation design explicitly addresses the user's perceived benefit from this sharing.

6. SUMMARY AND CONCLUSIONS

As a mechanism to improve usability, system-created recommendations have been a part of end-user development systems for a very long time. Unfortunately, few systems today take advantage of recommender technologies to cope with the increased scope of recommendation within EUD. Instead, handcrafted rules limit the types of suggestions that today's systems can make. As such, usability and adoption suffer.

We identify four main classes of recommendation that could be improved by using recommender technology:

- Inserting automation into the user's workflow
- Helping the user make the right decisions
- Handling errors
- Supporting unplanned sharing

Each of these classes of recommendation could make appearances in a variety of EUD systems, and as a whole they address user needs across the entirety of an end-user development workflow, from conceiving of a needed customization, to programming it, to sharing it with others.

In the space of inserting automation, recommenders could improve suggestions over shared repositories and perhaps directly improve activity recognition algorithms. Here, the major research challenges involve incorporating the user's operating environment into the recommendation algorithm to make the recommendations appropriately context-aware.

For improving user decision making, recommenders can help to organize the decision space, providing sensible defaults for a number of programming decisions, such as performing dataflow changes in an editor or choosing a best generalization in a PBD system. Challenges here include making recommendations about information-sparse decisions like simple edits. Again, context awareness could help to solve this problem.

When handling errors, opportunities for recommender systems are numerous, including suggesting potential problems, code examples, and possible fixes. While suggesting code examples is a fairly straightforward recommendation application, identifying potential problems and fixes exposes recommender algorithms to a nontraditional set of features to reason over. Learning exactly what features allow recommenders to identify similar programs is an interesting challenge we pose to the community.

Finally, supporting unplanned sharing allows recommender systems to apply their ability to leverage communities to the world of end-user development. Systems could help users to identify reusable components, especially aiding novices. Further, automatic identification of best practices could improve performance even for advanced end-user developers.

In all of these cases, end-user development provides a natural environment in which to evaluate recommendations in the context of real user workflows. Many evaluations in the end-user development space already combine user-centered evaluation methods with algorithm performance evaluation; user-centric evaluation of recommenders in this space simply requires evaluators to extend existing protocol designs to isolate the features unique to recommenders.

In conclusion, EUD systems have just begun to scratch the surface of how recommendations can improve user experience. By increasing communication between the EUD and recommender communities and creating recommendations that meet real user needs, we can move development out of the realm of the specialist and into the real world.

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA) or the Air Force Research Laboratory (AFRL).

Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, Sep 30, 2010
Published by CEUR-WS.org, ISSN 1613-0073, online ceur-ws.org/Vol-612/paper7.pdf

8. REFERENCES

[1] Adomavicius, G., Sankaranarayanan, R., Sen, S. and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. on Information Systems*. 23, 1 (2005), 103–145.

[2] Basili, V.R., Briand, L.C. and Melo, W.L. 1996. How reuse influences productivity in object-oriented systems. *Communications of the ACM*. 39, 10 (1996), 116.

[3] Boehm, B.W., Madachy, R. and Steece, B. 2000. *Software Cost Estimation with Cocomo II*. Prentice Hall, NJ.

[4] Brandt, J., Guo, P., Lewenstein, J., Dontcheva, M. and Klemmer, S. 2009. Two studies of opportunistic programming: Interleaving web foraging, learning, and writing code. *Proc. 27th Conference on Human Factors in Computing Systems* (2009), 1589–1598.

[5] Cypher, A. and Halbert, D.C. 1993. *Watch What I Do: Programming by Demonstration*. MIT press.

[6] Cypher, A. 1991. EAGER: Programming repetitive tasks by example. *Proc. 9th Conference on Human Factors in Computing Systems* (New Orleans, LA, 1991), 33-39.

[7] Garvey, T., Gervasio, M., Lee, T., Myers, K., Angiolillo, C., Gaston, M., Knittel, J. and Kolojejchick, J. 2009. Learning by demonstration to support military planning and decision making. *Proc. 21st Conference on Applications of Artificial Intelligence* (2009).

[8] Gervasio, M., Lee, T.J. and Eker, S. Learning email procedures for the desktop. *Proc. AAAI 2008 Workshop on Enhanced Messaging*.

[9] Gervasio, M. and Murdock, J. 2009. What were you thinking? Filling in missing dataflow through inference in learning from demonstration. *Proc. 14th Conference on Intelligent User Interfaces* (2009).

[10] Gilmore, D.J. 1991. Models of debugging. *Acta Psychologica*. 78, 1-3 (1991), 151–172.

[11] Haines, W., Gervasio, M., Blythe, J., Lerman, K. and Spaulding, A. 2010. A world wider than the web: End user programming across multiple domains. *No Code Required*. Morgan Kaufmann. 213-231.

[12] Hofmann, T. 2003. Collaborative filtering via Gaussian probabilistic latent semantic analysis. *Proc. 26th Conference on Research and Development in Informaion Retrieval* (2003), 266.

[13] Horvitz, E., Breese, J., Heckerman, D., Hovel, D. and Rommelse, K. 1998. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence* (1998), 256–265.

[14] Ko, A.J., DeLine, R. and Venolia, G. 2007. Information needs in collocated software development teams. *Proc. 29th Conference on Software Engineering* (2007), 344–353.

[15] Ko, A.J. and Myers, B.A. 2004. Designing the whyline: A debugging interface for asking questions about program behavior. *Proc. 22nd Conference on Human Factors in Computing Systems* (2004), 158.

[16] Ko, A.J. and Myers, B.A. 2008. Debugging reinvented. *Proc. 30th Conference on Software Engineering* (2008), 301–310.

[17] Ko, A.J. and Myers, B.A. 2009. Finding causes of program output with the Java Whyline. *Proc. 27th Conference on Human Factors in Computing Systems* (2009), 1569–1578.

[18] Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwing, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M. and Wiedenbeck, S. 2010. The state of the art in end-user software engineering. *ACM Computing Surveys*. (2010).

[19] Landauer, T.K. and Dumais, S.T. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*. 104, 2 (1997), 211–240.

[20] Leshed, G., Haber, E.M., Matthews, T. and Lau, T. 2008. CoScripter: Automating & sharing how-to knowledge in the enterprise. *Proc. 26th Conference on Human Factors in Computing Systems*. (2008).

[21] Lieberman, H. 2001. *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, San Francisco, CA.

[22] Lieberman, H., Paterno, F., Klann, M. and Wulf, V. 2006. End-user development: An emerging paradigm. *End User Development*. (2006), 1–8.

[23] Mackay, W.E. 1990. Patterns of sharing customizable software. *Proc. Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, 1990), 209-221.

[24] Masui, T. and Nakayama, K. 1994. Repeat and predict: Two keys to efficient text editing. *Proc. 12th Conference on Human Factors in Computing* (1994), 118–130.

[25] Nardi, B.A. 1993. *A small matter of programming: Perspectives on end user computing*. The MIT Press.

[26] Nielsen, J. and Molich, R. 1990. Heuristic evaluation of user interfaces. *Proc. 8th Conference on Human Factors in Computing Systems* (1990), 249-256.

[27] Peintner, B., Dinger, J., Rodriguez, A. and Myers, K. 2009. Task assistant: Personalized task management for military environments. *IAAI-09*. (2009).

[28] Ruvini, J. and Dony, C. 2000. APE: Learning user's habits to automate repetitive tasks. *Proc. 5th Conference on Intelligent User Interfaces* (2000), 229–232.

[29] Tassey, G. 2002. The economic impacts of inadequate infrastructure for software testing. *National Institute of Standards and Technology, RTI Project*. (2002).

[30] Wolfman, S., Lau, T., Domingos, P. and Weld, D. 2001. Mixed initiative interfaces for learning tasks: SMARTedit talks back. *Proc. 6th Conference on Intelligent User Interfaces* (2001), 167–174.

[31] Wright, P.C. and Monk, A.F. 1991. The use of think-aloud evaluation methods in design. *SIGCHI Bull.* 23, 1 (1991), 55-57.

[32] Ye, Y. and Fischer, G. 2005. Reuse-conducive development environments. *Automated Software Engineering*. 12, 2 (2005), 199–235.

[33] Yorke-Smith, N., Saadati, S., Myers, K.L. and Morley, D.N. 2009. Like an intuitive and courteous butler: A proactive personal agent for task management. *Proc. 8th Conference on Autonomous Agents and Multiagent Systems-Volume 1* (2009), 337–344.