# Making Recommendations Better: An Analytic Model for Human-Recommender Interaction

**Sean M. McNee**
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
mcnee@cs.umn.edu

**John Riedl**
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
riedl@cs.umn.edu

**Joseph A. Konstan**
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
konstan@cs.umn.edu


CHI 2006, April 22–27, 2006, Montreal, Canada.
ACM 1-xxxxxxxxxxxxxxxxxx.

## Abstract
Recommender systems do not always generate good recommendations for users. In order to improve recommender quality, we argue that recommenders need a deeper understanding of users and their information seeking tasks. Human-Recommender Interaction (HRI) provides a framework and a methodology for understanding users, their tasks, and recommender algorithms using a common language. Further, by using an analytic process model, HRI becomes not only descriptive, but also constructive. It can help with the design and structure of a recommender system, and it can act as a bridge between user information seeking tasks and recommender algorithms.

## Keywords
Human-Recommender Interaction, personalization, recommender systems, collaborative filtering, user-centered design, information seeking

## ACM Classification Keywords
H3.3. Information Search and Retrieval: Information Filtering, Search Process, Retrieval Models

## Introduction

In 2002, Jeffery Zaslow of the Wall Street Journal wrote an article about recommender systems entitled “If TiVo Thinks You Are Gay, Here's How to Set It Straight”. He talked of TiVo, Amazon.com, and NetFlix, among others generating recommendations that made absolutely no sense to the users of these systems. The article stated that users felt the recommenders didn’t understand them, and, more importantly, the recommenders had *their own opinions*. It ends with a telling quotation, “Mr. Leon believes the box was giving them a message: ‘You're definitely gay. And you're watching too much TV.’” [Zaslow 2002].

While this generated a large amount of press for personalization and recommenders, they weren’t presented in the most positive light. As recommender systems researchers, there is a lot we can learn from feedback like this. We will focus on two points:

1. Recommenders don’t understand *why* a user wants recommendations.

2. Users interact with the recommenders in a conversational style. To them, recommenders have personalities.

Historically, research has focused on making recommendations more accurate, with the implicit assumption that ‘more accurate equates to ‘better liked and more useful’. Lately, researchers have been questioning this assumption [Herlocker 2004]. As a community, perhaps, we’ve gotten it backwards. First we need to determine what kinds of recommendations are good and useful to users. Next we should devise the appropriate metrics and test algorithm performance. We need to re-think about recommenders from a user-centric perspective.

We’ll start with the question, “Why do users come to recommenders?” We postulate that users have specific information needs and come to a recommender as part of an information seeking process. Assuming this, the question then becomes: “How can we, as recommender system designers and researchers, create recommenders that can generate useful recommendation lists for users with differing information needs?”

Our previous work [McNee 2002, Torres 2004, Ziegler 2005] has given us a key observation: While most well known recommender algorithms score similarly on accuracy metrics, these algorithms generate qualitatively different recommendation *lists*. Users, unlike most researchers, were not judging the quality of individual recommendations, but were instead experiencing the list as a whole, considering properties such as how well the breadth or depth of the list met their goals. Most important, end users noticed these qualitative differences, and in some cases preferred lists of recommendations that were less accurate (by traditional measures) but more suitable for their needs.

## An Example User and his Task

Looking in the domain of recommending research papers in a digital library environment, we could imagine a user like this one:

Max is a new graduate student writing one of his first conference papers. He has done great research and carried out an exhaustive search for related work. He still feels however, that he could be missing an

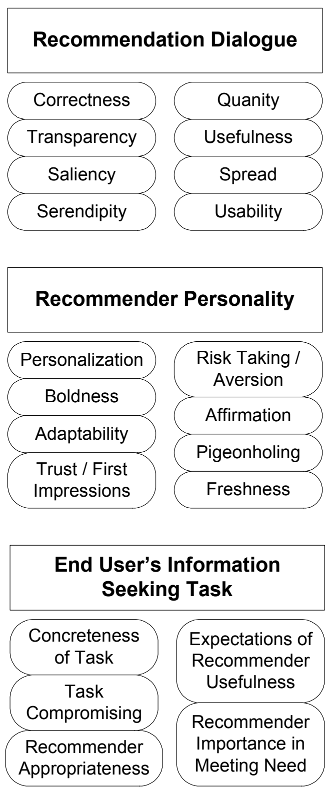


**Figure 1**: There are three Pillars to Human-Recommender Interaction. Each one is contains several aspects.

important citation or perhaps may be citing the 'wrong paper' for an area. He's looking for validation of the related work he's already found and suggestions on what areas he might have missed.

There are several ways to analyze Max's situation. We propose the following: We would say that Max is a novice in his field. To us, this would mean he is concerned about trusting the recommender, if he understands the recommendations, and whether or not he is even looking for the right information. We would also state his user task as "Filling out a reference list" and he would be concerned about how results would change based on his input, that the results made sense to him (felt good to him), and that he'd looked through the whole database for possible results.

This extra information about Max and his situation could dramatically change the kinds of paper recommendations he receives from a recommender. Human-Recommender Interaction theory (HRI) is a way to incorporate this task knowledge into the recommender system. We can use HRI to understand user needs in a recommender, and we can use the HRI Process Model to tailor the recommender to meet these needs.

## Human-Recommender Interaction

Human-Recommender Interaction (HRI) is a framework and methodology for analyzing user tasks and recommender algorithms with the end goal of generating useful recommendation lists. It was developed by re-examining the recommendation process from an end user's perspective: HRI categorizes aspects of the interactions a user has with a recommender based on user experiences and expectations developed over a period of time. It is a language to describe the kinds of recommendations which would help solve a user's information need.

As shown in Figure 1, there are three Pillars to HRI: The Recommendation Dialog, the act of giving information and receiving one recommendation list from a recommender; The Recommender Personality, the user's perception of the recommender over a period of time; and the User Information Seeking Tasks, the reason the user came to the recommender system. Each pillar contains various relevant aspects to that part of the interaction. A user's needs and expectations from a recommender can be described by selecting the most relevant aspects from each pillar.

*The Recommendation Dialog*
*Correctness* is judgment of the user that a recommendation provided by the recommender is a good and high quality recommendation for his information need. A *useful* recommendation is one that is going to be consumed by a user and/or help the user with his information seeking task. A correct recommendation may not be useful (i.e. the user already knows this item). A *transparent* recommendation is one that the user understands why it was recommended for her particular information task.

*Salience* means to "stand out". It is vivid, unexpected, notable, conspicuous, and prominent. A salient recommendation generates an emotional response from the target user, either positive or negative. In a related fashion, a *serendipitous* recommendation is an unexpected and fortunate recommendation. There is a strong relationship between salience and serendipity, but it is possible to have one without the other. We argue that the goal of a recommender is to generate salient recommendations—recommendations that strike an emotional response from a user—but not necessarily serendipitous ones.

The *quantity* of items returned in a recommendation list can affect the user's opinion of a system, as can the perceived *spread* of the items recommended—the user's perception of the percentage of items in the domain considered for this recommendation list. Finally, the *usability* of the recommender's interface plays an important role in creating a smooth dialog.

*The Recommender Personality*

*Personalization* is how different the recommendations are for each user of the recommender system. Related, the *adaptability* of the system is how the recommendations change in response to changes in a user's profile. For example, while a user might feel that the system is personalized to him, he could think that it doesn't adapt to subtle changes in his opinions. *Freshness* refers to how well the system can show new recommendations each time the user interacts with the system for the same or a similar task.

How strongly the recommender recommends particular items to users is the *boldness* of the recommender. A bold recommender could, for example, make many recommendations at the extremes of the prediction scale. *Risk* is related to boldness; it is the perception of the recommender to consistently generate recommendations for obscure, under-represented, or possibly unrelated items.

*Trust* can be difficult to establish and maintain in a recommender; many opinions are formed based on a user's *first impressions* of the system. A recommender can be *affirming* by showing items the user is more likely to be aware of; such safe environments can be used to establish rapport with new users. This can be complicated because a user could feel pigeonholed—that she only gets recommendations of a small subset of all possible items in the system. It is worth noting that both affirmation and pigeonholing can be used for various purposes: pigeonholing is good when a user has a focused need, and affirmation is good when an experienced user has an unfamiliar task.

*The User Information Seeking Task*

We believe users have a reason for coming to a recommender. They have information needs. A user may or may not be able to express his need verbally; we describe this 'fuzziness' as the *concreteness* of the task. In a related fashion, a user may change their task as they use the system, when the do their original task becomes *compromised*. Recommenders should not expect users always to have concrete tasks and realize that many tasks will be compromised.

Existing users will make decisions about whether or not to return to a recommender to help with a new task; a recommender may play a small *role in meeting their need*. A recommender may be of high quality and frequented by the user, but that doesn't mean that it is *appropriate* for any given information need. Finally, if a user does choose to use a recommender to help meet a need, the user will have *expectations of usefulness* of this recommender before she even starts using it. Understanding this context will help a recommender when a user does choose to use it [Case 2002].

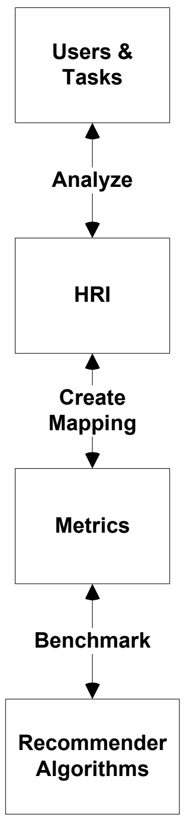


**Figure 2**: The HRI Analytic Process Model. Interactions flow two ways in this model, linking recommender algorithms to users and their information seeking tasks via HRI aspects.

## The HRI Analytic Process Model

By itself, Human-Recommender Interaction is a descriptive model, a way to understand recommenders from a different point of view. When added to a larger process model however, it becomes constructive—a way to analyze and redesign recommenders to better meet user information needs. Figure 2 shows an overview of the HRI Analytic Process Model.

A user interacts with a recommender as part of an information seeking process. More importantly, the user establishes a relationship with a recommender through repeated information dialogs. As such, a key part of this model is to understand the user information seeking tasks in the recommendation domain. For example, why would a user come to a book recommender? To buy a book? To find a book for a gift? To read for pleasure? For professional development? To study for a test?

As the information density of a domain increases, users will have more specific needs (e.g. finding research papers instead of finding movies) [Im 2001]. The language we use to describe these tasks is HRI. First we need a list of user types and typical domain tasks. There are a variety of ways to gather this information (e.g. [Hackos 1998]). A detailed analysis of these tasks will allow us to link tasks to specific HRI Aspects. For example, would a particular task require risky recommendations or perhaps recommendations from a more affirming recommender? Only some aspects will be important to any one given task.

By looking at which HRI aspects are important to different user tasks, we can design metrics to categorize the relevant and important differences between tasks. For example, spread and pigeonholing suggest the importance of a metric to study item similarity in a recommendation list, such as the Intra-List Similarity Metric proposed in [Ziegler 2005].

These metrics can then, in turn, benchmark a wide variety of known recommender algorithms—a catalog of algorithm behaviors. We believe that several different metrics are needed to capture the differences that accuracy metrics alone cannot gather.

## Back to Our Example, Max

Max was a novice user looking to fill out references for a research paper. After reviewing HRI, we would say his relevant HRI Aspects are: Saliency, Serendipity, Quantity, Spread, Adaptability, Trust, Task Concreteness, and Recommender Appropriateness.

These are only a selection of possible aspects for Max. It is important to choose carefully and apply the same selection process across all user tasks for this domain. Aspects are not fixed either; we expect HRI itself to grow and change as it is applied to more domains.

## Implications, Discussion, and Future Work

HRI is a framework and a language for understanding and describing users, their information needs, and recommender components. It allows designers to reflect on system aspects that they may have not previously considered (e.g. What kinds of users do we want to target? How risky is our recommender?).

We can use the results of the Analytic Model to view the information space to understand what a recommender can and cannot do for a particular domain. By seeing which HRI aspects are most important to users of a domain, a recommender can adjust to meet those needs.

Through HRI and the Analytic Process Model, we can analyze user tasks, select their relevant aspects, and via a variety of metrics, we can link these tasks to recommender algorithms. We assert that certain algorithms are better suited to specific tasks, and that a recommender will need to have a family of algorithms at its disposal to select from when generating recommendations for users. This is a departure from the current model of 'one recommender for everyone' common in recommender systems today.

Is HRI useful? To test this, we designed experiments for all sections of the Process Model. We created a series of metrics and perform simulation experiments against well-known recommender algorithms. We performed a task analysis in the domain of computer science research papers to study the mapping between user tasks and recommender algorithms via HRI aspects [all submitted, under review]. Finally, we will run user studies to determine task-to-HRI mappings and validate the usefulness of our approach.

## Conclusion

Recommender systems need a deeper understanding of users and their information seeking tasks to be able to generate better recommendations. HRI provides a new user-centric view of recommender systems. Through HRI, system designers and researchers have a common language to explore how users perceive and user recommenders. The Analytic Process model goes one step further and allows us to use HRI not just to talk about recommenders, but to analyze, categorize, and redesign them to better meet user information needs.

## Acknowledgements

We would like to thank Cai-Nicolas Ziegler, Roberto Torres, Loren Terveen, and all of GroupLens Research. This work is funded by the NSF, grants DGE 95-54517, IIS 96-13960, 97-34442, 99-78717, and 01-02229.

## References

[1] Case, D.O. *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior*. Academic Press: San Diego, CA, 2002.

[2] Hackos, J.T., Redish, J.C. *User and Task Analysis for Interface Design*. Wiley, Hoboken, NJ, 1998.

[3] Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J.T. Evaluating Collaborative Filtering Recommender Systems. *ACM TOIS* 22, 1 (2004), 5-53.

[4] Im, I., Hars, A. Finding Information Just For You: Knowledge Reuse Using Collaborartive Filtering Systems. In *Proc. of ICIS 2001*, Association for Information Systems (2001), 349-360.

[5] 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 CSCW 2002*, ACM Press (2002), 116-125.

[6] Torres, R., McNee, S.M., Abel, M., Konstan, J.A., and Riedl, J. Enhancing digital libraries with TechLens$^{+}$. In *Proc. of JCDL 2004*, ACM Press (2004) 228-236.

[7] Zaslow, J. If TiVo Thinks You Are Gay, Here's How To Set It Straight -- Amazon.com Knows You, Too, Based on What You Buy; Why All the Cartoons? The Wall Street Journal, sect. A, p. 1, November 26, 2002.

[8] 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.