Tell Me More? The Effects of Mental Model Soundness on Personalizing an Intelligent Agent

Todd Kulesza[1], Simone Stumpf[2], Margaret Burnett[1], Irwin Kwan[1]

[1]Oregon State University
School of EECS
Corvallis, Oregon 97333
{kuleszto, burnett, kwan}@eecs.oregonstate.edu

[2]City University London
Centre for HCI Design, School of Informatics
London EC1V 0HB, United Kingdom
Simone.Stumpf.1@city.ac.uk

ABSTRACT

What does a user need to know to productively work with an intelligent agent? Intelligent agents and recommender systems are gaining widespread use, potentially creating a need for end users to understand how these systems operate in order to fix their agent's personalized behavior. This paper explores the effects of mental model soundness on such personalization by providing structural knowledge of a music recommender system in an empirical study. Our findings show that participants were able to quickly build sound mental models of the recommender system's reasoning, and that participants who most improved their mental models during the study were significantly more likely to make the recommender operate to their satisfaction. These results suggest that by helping end users understand a system's reasoning, intelligent agents may elicit more and better feedback, thus more closely aligning their output with each user's intentions.

Author Keywords

Recommenders; mental models; debugging; music; personalization; intelligent agents;

ACM Classification Keywords

H.5.m [Information interfaces and presentation]: Miscellaneous;

INTRODUCTION

Intelligent agents have moved beyond mundane tasks like filtering junk email. Search engines now exploit pattern recognition to detect image content (e.g., clipart, photography, and faces); Facebook and image editors take this a step further, making educated guesses as to who is in a particular photo. Netflix and Amazon use collaborative filtering to recommend items of interest to their customers, while Pandora and Last.fm use similar techniques to create radio stations crafted to an individual's idiosyncratic tastes. Simple rule-based systems have evolved into agents employing complex algorithms. These *intelligent agents* are computer programs whose behavior only becomes fully specified *after* they learn from an end user's training data.

Because of this period of in-the-field learning, when an intelligent agent's reasoning causes it to perform incorrectly or unexpectedly, only the end user is in a position to better personalize—or more accurately, to debug—the agent's flawed reasoning. Debugging, in this context, refers to *mindfully and purposely* adjusting the agent's reasoning (after its initial training) so that it more closely matches the user's expectations. Recent research has made inroads into supporting this type of functionality [1,11,14,16]. Debugging, however, can be difficult for even trained software developers—helping end users do so, when they lack knowledge of either software engineering or machine learning, is no trivial task.

In this paper, we consider how much ordinary end users may need to know about these agents in order to debug them. Prior work has focused on how an intelligent agent can explain itself to end users [9,13,15,22,27,28], and how end users might act upon such explanations to debug their intelligent agents [1,11,14,16,24]. This paper, in contrast, considers whether users actually need a sound mental model, and how that mental model impacts their attempts to debug an intelligent agent. Toward this end, we investigated four research questions:

(RQ1): Feasibility: Can end users quickly build and recall a sound mental model of an intelligent agent's operation?

(RQ2): Accuracy: Do end users' mental models have a positive effect on their debugging of an intelligent agent?

(RQ3): Confidence: Does building a sound mental model of an intelligent agent improve end users' computer self-efficacy and reduce computer anxiety?

(RQ4): User Experience: Do end users with sound mental models of an intelligent agent experience interactions with it differently than users with unsound models?

To answer these research questions, we conducted an empirical study that investigates the effects of explaining the reasoning of a music recommender system to end users. We developed a prototype, *AuPair*, which allowed participants to set up radio stations and make adjustments to

CHI'12, May 5–10, 2012, Austin, Texas, USA.

the songs that it chose for them. Half of the participants received detailed explanations of the recommender's reasoning, while the other half did not. Our paper's contribution is a better understanding of how users' mental models of their intelligent agents' behavior impacts their ability to debug their personalized agents.

BACKGROUND AND RELATED WORK

Functional and Structural Mental Models

Mental models are internal representations that people build based on their experiences in the real world. These models allow people to understand, explain and predict phenomena, and then act accordingly [10]. The contents of mental models can be concepts, relationships between concepts or events (e.g., causal, spatial, or temporal relationships), and associated procedures. For example, one mental model of how a computer works could be that it simply displays everything typed on the keyboard and "remembers" these things somewhere inside the computer's casing. Mental models can vary in their richness—an IT professional, for instance, has (ideally) a much richer mental model of how a computer works.

There are two main kinds of mental models: *Functional* (shallow) models imply that the end user knows how to use the computer but not how it works in detail, whereas *structural* (deep) models provide a detailed understanding of how and why it works. Mental models must be sound (i.e., accurate) enough to support effective interactions; many instances of unsound mental models guiding erroneous behavior have been observed [18].

Mental model completeness can matter too, especially when things go wrong, and structural models are more complete than functional models. While a structural model can help someone deal with unexpected behavior and fix the problem, a purely functional model does not provide the abstract concepts that may be required [10]. Knowing how to *use* a computer, for example, does not mean you can *fix* one that fails to power on.

To build new mental models, it has been argued that users should be exposed to transparent systems and appropriate instructions [21]. *Scaffolded instruction* is one method that has been shown to contribute positively to learning to use a new system [20]. One challenge, however, is that mental models, once built, can be surprisingly hard to shift, even when people are aware of contradictory evidence [28].

Mental Models of an Intelligent Agent's Reasoning

There has been recent interest in supporting the debugging of intelligent agents' reasoning [1,11,13,14,16,25], but the mental models users build while attempting this task have received little attention. An exception is a study that considered the correctness of users' mental models when interacting with a sensor-based intelligent agent that predicted an office worker's availability (e.g., "Is now a good time to interrupt so-and-so?") [28], but this study did not allow users to debug these availability predictions.

Making an agents' reasoning more transparent is one way to influence mental models. Examples of explanations by the agent for specific decisions include *why…* and *why not…* descriptions of the agent's reasoning [13,15], visual depictions of the assistant's known correct predictions versus its known failures [26], and electronic "door tags" displaying predictions of worker interruptibility with the reasons underlying each prediction (e.g., "talking detected") [28]. Recent work by Lim and Dey has resulted in a toolkit for applications to generate explanations for popular machine learning systems [16]. Previous work has found that users may change their mental models of an intelligent agent when the agent makes its reasoning transparent [14]; however, some explanations by agents may lead to only shallow mental models [24]. Agent reasoning can also be made transparent via explicit instruction regarding new features of an intelligent agent, and this can help with the construction of mental models of how it operates [17]. None of these studies, however, investigated how mental model construction may impact the ways in which end users debug intelligent agents.

Making an intelligent agent's reasoning transparent can improve perceptions of satisfaction and reliability toward music recommendations [22], as well as other types of recommender systems [9,27]. However, experienced users' satisfaction may actually decrease as a result of more transparency [17]. As with research on the construction of mental models, these studies have not investigated the link between end users' mental models and their satisfaction with the intelligent agent's behavior.

EMPIRICAL STUDY

To explore the effects of mental model soundness on end-user debugging of intelligent agents, we needed a domain that participants would be motivated to both use and debug. Music recommendations, in the form of an adaptable Internet radio station, meet these requirements, so we created an Internet radio platform (named *AuPair*) that users could personalize to play music fitting their particular tastes.

To match real-world situations in which intelligent agents are used, we extended the length of our empirical study beyond a brief laboratory experiment by combining a controlled tutorial session with an uncontrolled period of field use. The study lasted five days, consisting of a tutorial session and pre-study questionnaires on Day 1, then three days during which participants could use the AuPair prototype as they wished, and an exit session on Day 5.

AuPair Radio

AuPair allows the user to create custom "stations" and personalize them to play a desired type of music. Users start a new station by seeding it with a single artist name (e.g., "Play music by artists similar to *Patti Smith*"). Users can debug the agent by giving feedback about individual songs, or by adding general guidelines to the station. Feedback about an individual song can be provided using the 5-point

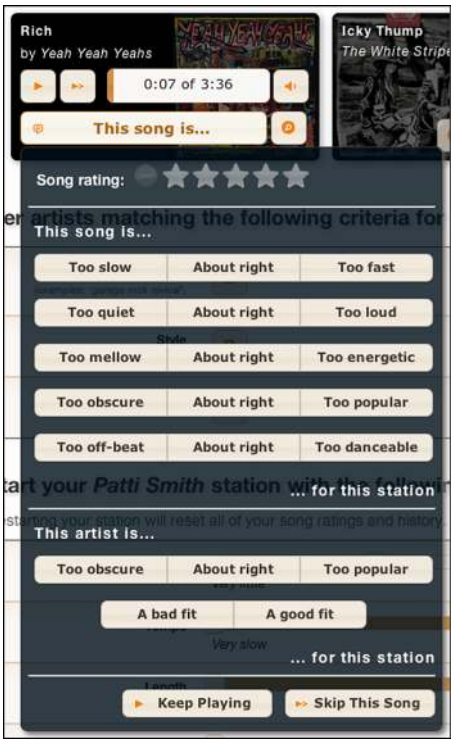

Figure 1. Users could debug by saying *why* the current song was a good or bad choice.

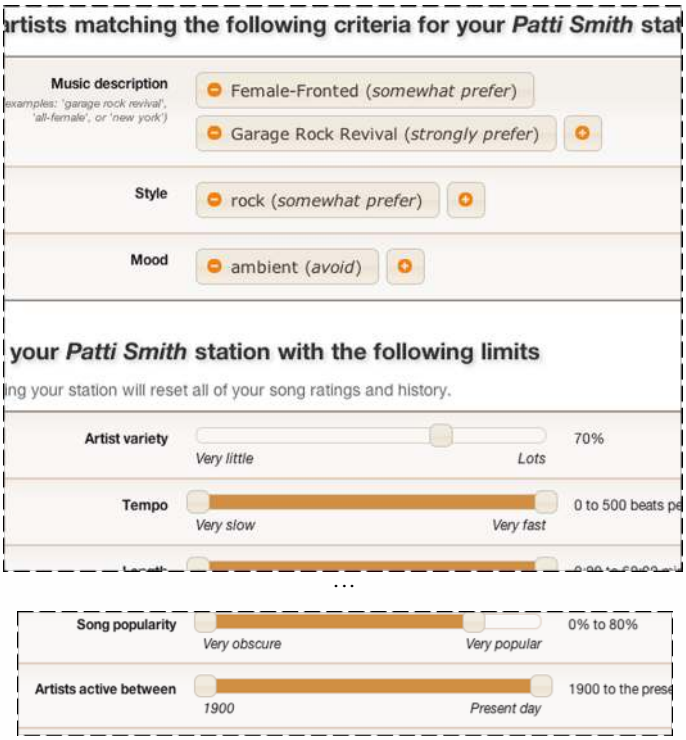

Figure 2. Participants could debug by adding guidelines on the type of music the station should or should not play, via a wide range of criteria.

rating scale common to many media recommenders, as well as by talking about the song's attributes (e.g., "This song is too mellow, play something more energetic", Figure 1). To add general guidelines about the station, the user can tell it to "prefer" or "avoid" descriptive words or phrases (e.g., "Strongly prefer garage rock artists", Figure 2, top). Users can also limit the station's search space (e.g., "Never play songs from the 1980's", Figure 2, bottom).

AuPair was implemented as an interactive web application, using jQuery and AJAX techniques for real-time feedback in response to user interactions and control over audio playback. We supported recent releases of all major web browsers. A remote web server provided recommendations based on the user's feedback and unobtrusively logged each user interaction via an AJAX call.

AuPair's recommendations were based on The Echo Nest [6], allowing access to a database of cultural characteristics (e.g., genre, mood, etc.) and acoustic characteristics (e.g., tempo, loudness, energy, etc.) of the music files in our library. We built our music library by combining the research team's personal music collections, resulting in a database of more than 36,000 songs from over 5,300 different artists.

The Echo Nest developer API includes a dynamic playlist feature, which we used as the core of our recommendation engine. Dynamic playlists are put together using machine learning approaches and are "steerable" by end users. This is achieved via an adaptive search algorithm that builds a path (i.e., a playlist) through a collection of similar artists. Artist similarity in AuPair was based on cultural characteristics, such as the terms used to describe the artist's music. The algorithm uses a clustering approach based on a distance metric to group similar artists, and then retrieves appropriate songs. The user can adjust the distance metric (and hence the clustering algorithm) by changing weights on specific terms, causing the search to prefer artists matching these terms. The opposite is also possible—the algorithm can be told to completely avoid undesirable terms. Users can impose a set of limits to exclude particular songs or artists from the search space. Each song or artist can be queried to reveal the computer's understanding of its acoustic and cultural characteristics, such as its tempo or "danceability".

Participants

Our study was completed by 62 participants, (29 females and 33 males), ranging in age from 18 to 35. Only one of the 62 reported prior familiarity with computer science. These participants were recruited from Oregon State University and the local community via e-mail to university students and staff, and fliers posted in public spaces around the city (coffee shops, bulletin boards, etc.). Participants were paid $40 for their time. Potential participants applied via a website that automatically checked for an HTML5-compliant web browser (applicants using older browsers were shown instructions for upgrading to a more recent

browser) to reduce the chance of recruiting participants who lacked reliable Internet access or whose preferred web browser would not be compatible with our prototype.

Experiment Design & Procedure

We randomly assigned participants to one of two groups—a *With-scaffolding* treatment group, in which participants received special training about AuPair's recommendation engine, and a *Without-scaffolding* control group. Upon arrival, participants answered a widely used, validated self-efficacy questionnaire [5] to measure their confidence in problem solving with a hypothetical (and unfamiliar) software application.

Both groups then received training about AuPair, which differed only in the depth of explanations of how AuPair worked. The Without-scaffolding group was given a 15-minute tutorial about the functionality of AuPair, such as how to create a station, how to stop and restart playback, and other basic usage information. The same researcher provided the tutorial to every participant, reading from a script for consistency. To account for differences in participant learning styles, the researcher presented the tutorial interactively, via a digital slideshow interleaved with demonstrations and hands-on participation.

The With-scaffolding group received a 30-minute tutorial about AuPair (15 minutes of which was identical to the Without-scaffolding group's training) that was designed to induce not only a functional mental model (as with the Without-scaffolding group), but also a *structural* mental model of the recommendation engine. This "behind the scenes" training included illustrated examples of how AuPair determines artist similarity, the types of acoustic features the recommender "knows" about, and how it extracts this information from audio files. Researchers systematically selected content for the scaffolding training by examining each possible user interaction with AuPair and then describing how the recommender responds. For instance, every participant was told that the computer will attempt to "play music by similar artists", but the With-scaffolding participants were then taught how *tf-idf* (term frequency-inverse document frequency, a common measure of word importance in information retrieval) was used to find "similar" artists. In another instance, every participant was shown a control for using descriptive words or phrases to steer the agent, but only With-scaffolding participants were told *where* these descriptions came from (traditional sources, like music charts, as well as Internet sources, such as Facebook pages).

After this introduction, each participant answered a set of six multiple-choice comprehension questions in order to establish the soundness of their mental models. Each question presented a scenario (e.g., "Suppose you want your station to play more music by artists similar to *The Beatles*"), and then asked which action, from a choice of four, would best align the station's recommendations with the stated goal. Because mental models are inherently "messy, sloppy… and indistinct" [18], we needed to determine if participants were guessing, or if their mental models were sound enough to eliminate some of the incorrect responses. Thus, as a measure of confidence, each question also asked how many of the choices could be eliminated before deciding on a final answer. A seventh question asked participants to rate their *overall* confidence in understanding the recommender on a 7-point scale.

The entire introductory session (including questionnaires) lasted 30 minutes for Without-scaffolding participants, and 45 minutes for With-scaffolding participants. Both groups received the same amount of hands-on interaction with the recommender.

Over the next five days, participants were free to access the web-based system as they pleased. We asked them to use AuPair for at least two hours during this period, and to create at least three different stations. Whenever a participant listened to music via AuPair, it logged usage statistics such as the amount of time they spent debugging the system, which debugging controls they used, and how frequently these controls were employed.

After five days, participants returned to answer a second set of questions. These included the same self-efficacy and comprehension questionnaires as on Day 1 (participants were not told whether their comprehension responses were correct), plus the NASA-TLX survey to measure perceived task load [8]. We also asked three Likert-scale questions about user's satisfaction with AuPair's recommendations, using a 21-point scale for consistency with the NASA-TLX survey, and the standard Microsoft Desirability Toolkit [3] to measure user attitudes toward AuPair.

Data Analysis

We used participants' answers to the comprehension questions described earlier to measure mental model soundness. Each question measured the depth of understanding for a specific type of end user debugging interaction, and their combination serves as a reasonable proxy for participants' understanding of the entire system. We calculated the soundness of participant's mental models using the formula $\sum_i (correctness_i \times confidence_i)$, where *correctness* is either 1 for a correct response, or -1 for an incorrect response and *confidence* is a value between 1 and 4 (representing the number of answers the participant was able to eliminate). These values were summed for each question i to create a participant's comprehension score, ranging from -24 (indicating a participant who was completely confident about each response, but always wrong) to +24 (indicating someone who was completely confident about each response and always correct).

Mental models evolve as people integrate new observations into their reasoning [18], and previous studies have suggested that participants may adjust their mental models while working with an intelligent agent that is transparent about its decision-making process [14]. Furthermore,

constructivist learning theory [12] places emphasis on knowledge *transformation* rather than the overall *state* of knowledge. Hence, we also calculated *mental model transformation* by taking the difference of participants' two comprehension scores (*day_5_score – day_1_score*). This measures how much each participant's knowledge shifted during the study, with a positive value indicating increasing soundness, and a negative value suggesting the replacement of sound models with unsound models.

Table 1 lists all of our metrics and their definitions.

RESULTS

Feasibility (RQ1)

Effectiveness of Scaffolding

Understanding how intelligent agents work is not trivial—even designers and builders of intelligent systems may have considerable difficulty [11]. Our first research question (RQ1) considers the feasibility of inducing a sound mental model of an algorithm's reasoning process in end users—if participants fail to learn how the recommender works given a human tutor in a focused environment, it seems unreasonable to expect them to learn it on their own.

We tested for a difference in mental model soundness (measured by comprehension scores weighted by confidence) between the With-scaffolding group and the Without-scaffolding group. The With-scaffolding group had significantly higher scores than the Without-scaffolding group, both before and after the experiment task (Day 1: Welch's t-test, p=.004, t=-3.03, df=53.64) (Day 5: Welch's t-test, p<.001, t=-3.77, df=59.87). To ensure these differences were not primarily the result of differing levels of confidence, we performed the same test without weighting the comprehension scores by confidence, finding nearly identical results (Day 1: Welch's t-test, p=.003, t=-3.09, df=55.11) (Day 5: Welch's t-test, p<.001, t=-3.55, df=59.36). Neither group's mean comprehension score changed significantly during the 5-day study (Figure 3).

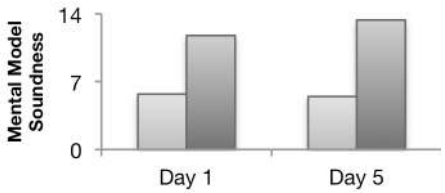

Figure 3. With-scaffolding participants (dark) held sounder mental models than without-scaffolding participants (light), both immediately following the tutorial, and five days later.

Participants also showed differences in their *perceived* mental model soundness, at least at first. On Day 1, the Without-scaffolding group was significantly less certain that they accurately understood how the system selected songs and responded to feedback (mean score of 4.5 out of 7) than the With-scaffolding group (mean score of 5.6 out of 7) (Welch's t-test, p=.015, t=-2.51, df=58.00). By Day 5, however, the Without-scaffolding group's responses had risen to a mean of 5.25, with no evidence of statistical difference against the With-scaffolding group (with a mean of 5.3).

Discussion

These results provide insights into four aspects of the practicality of end users comprehending and debugging the reasoning of an intelligent agent.

First, even a short 15-minute scaffolding tutorial effectively taught participants how the recommender "reasoned". With-scaffolding participants were significantly more likely to correctly and confidently answer the comprehension questions. This in turn suggests that the With-scaffolding participants should be better equipped to debug the recommender's reasoning than the Without-scaffolding participants, a point we investigate in RQ2.

Second, mental model soundness did not significantly improve during the five days participants interacted with AuPair on their own—simply using the system did not significantly help participants develop sounder mental

Metric	Definition
Mental model soundness	Responses to comprehension questions (sum of correct responses, weighted by confidence).
Perceived mental model soundness	Response to Likert question "Are you confident all of your statements are accurate?" after participants were asked to enumerate how they think the recommender made decisions.
Mental model transformation	Post-task mental model soundness minus pre-task mental model soundness.
Debugging interactions	Number of actions a participant used to debug the playlist (e.g., providing feedback, getting the next recommendation, or viewing a song's features), from the automated log files.
Interaction time	Length of time a participant spent on the task, i.e. listening to and interacting with *AuPair*.
Cost/benefit	Response to Likert question "Do you feel the effort you put into adjusting the computer was worth the result?"
Satisfaction	Response to Likert question "How satisfied are you with the computer's playlists?"

Table 1: Definitions for each metric used in our data analysis.

models about its reasoning. This is in contrast to recent work in interactive machine learning, which has found that for some systems (e.g., gesture recognition frameworks), repeated use taught people the most salient aspects of how the system worked [7].

Third, the soundness of participants' mental models largely persisted for the duration of the study. This appeared to be the case for both the Without-scaffolding and With-scaffolding groups, with neither groups' comprehension scores significantly changing between Day 1 and Day 5. This bodes well for end users retaining and recalling sound models initially learned about an intelligent agent.

Fourth, however, is the issue of initially building unsound models: once incorrect models were built, they were hard to shift. Even though the Without-scaffolding group formed less sound mental models, their confidence in their mental models increased, suggesting that they had convinced themselves they were, in fact, correct. Making *in situ* explanations available on an ongoing basis, such as in [9,14,26], may be a way to address this issue.

Together, these findings provide evidence that furnishing end users with a brief explanation on the structure of an intelligent agents' reasoning, such as the attributes used, how such attributes are collected, and the decision-making procedure employed, can significantly improve their mental model's soundness.

Accuracy (RQ2)

A recommender's effectiveness is in the eye of the beholder. Personalized recommendations cannot have a "gold standard" to measure accuracy—only the end users themselves can judge how well an agent's recommendations match their personal tastes. Hence, for our second research question (RQ2), we turned to a pair of more appropriate measures to explore the effects of mental model soundness on "accuracy"—cost/benefit and participant satisfaction.

Cost/Benefit

In theory, a sound mental model enables a person to reason effectively about their best course of action in a given situation [10]. Thus, we expected participants with sounder mental models (the With-scaffolding participants, according to the RQ1 results) to debug more effectively than those with less sound models. For example, knowing that the recommender could be steered more effectively by using unique, highly specific words (e.g., "Merseybeat") rather than broad, common descriptors (e.g., "oldies") should have helped such participants debug the agent's reasoning more effectively than participants who did not understand this.

Surprisingly, when using participants' perceptions of cost/benefit as a surrogate for effectiveness, the soundness of participants' mental models showed little impact on this measure of debugging effectiveness. However, mental model *transformation* was tied with cost/benefit: participants who most improved the soundness of their mental models reported that the effort of debugging was significantly more worthwhile than participants whose mental models improved less, or not at all (Table 2, row 1 & Figure 4A).

Participants' opinions of effectiveness were confirmed by their debugging interactions to adjust or assess AuPair's recommendations (e.g., providing feedback, getting the next recommendation, or viewing a song's features). The count of these debugging interactions was significantly correlated with the improvement in mental model soundness for With-scaffolding participants, while no such correlation existed among Without-scaffolding participants (Table 2, rows 2 and 3 & Figure 4B). *Sounder changes* to the mental model, then, may have had a positive effect on debugging, whereas changes in an initially unsound model did not serve the Without-scaffolding participants as well.

Further, participants who most improved the soundness of their mental models spent significantly less *time* on their interactions than others (Table 2, row 4 & Figure 4C). In light of the increases in perceived cost/benefit and debugging interactions, this suggests positive mental model transformations were linked to more efficient debugging.

An alternative explanation of the above results is that debugging interactions were responsible for participants' mental model transformations, rather than the other way around. Recall, however, that the Without-scaffolding group showed no correlation between debugging interactions and mental models (Table 2, row 3). Thus, the evidence suggests that it was the *in situ enhancement* of relatively *sound* models that was linked to improved attitudes toward debugging.

Satisfaction

Our second measure of debugging effectiveness and the accuracy of the result was participants' satisfaction with AuPair's resulting recommendations. To measure this, we asked participants (using a Likert scale) "How satisfied are you with the computer's playlists?" at the end of the study.

As with the cost/benefit results, neither treatment nor mental model soundness was predictive of participant satisfaction (Table 2, rows 5 and 6). However, here again, transformation of mental models appeared to matter—mental model transformation was marginally predictive of how satisfied participants felt with AuPair's playlists (Table 2, row 7). For example, the participant whose mental model's soundness decreased the most expressed dissatisfaction and a feeling of being unable to control the computer:

"The idea is great to be able to 'set my preferences', but if the computer continues to play what I would call BAD musical choices—I'd prefer the predictability of using Pandora."

	Metric	Statistical Test	Result	Figure
1	Mental model transformation vs. cost/benefit	Linear regression	p=.041, R^2=.07, F(1,60)=4.37	Figure 4A
2	Mental model transformation (With-scaffolding) vs. debugging interactions	Pearson correlation	p=.031, r=.39, t=2.27, df=28	Figure 4B
3	Mental model transformation (Without-scaffolding) vs. debugging interactions	Pearson correlation	p=.952, r=.01, t=0.06, df=30	
4	Mental model transformation vs. interaction time	Pearson correlation	p=.032, r=-.27, t=-2.19, df=60	Figure 4C
5	Satisfaction between With-scaffolding/Without-scaffolding groups	Welch's t-test	p=.129, t=1.53, df=59.9	
6	Satisfaction vs. mental model soundness	Linear regression	p=.272, R^2=.02, F(1,60)=1.23	
7	Satisfaction vs. mental model transformation	Linear regression	p=.053, R^2=.06, F(1,60)=3.89	
8	Satisfaction vs. cost/benefit	Pearson correlation	p<.001, r=.73, t=8.25, df=60	Figure 4D
9	Satisfaction vs. debugging interactions	Pearson correlation	p=.293, r=-.13, t=-1.06, df=60	

Table 2. Positive mental model transformations were consistently associated with better benefits, lower costs, and improved satisfaction (significant results shaded). Definitions for each metric are listed in Table 1.

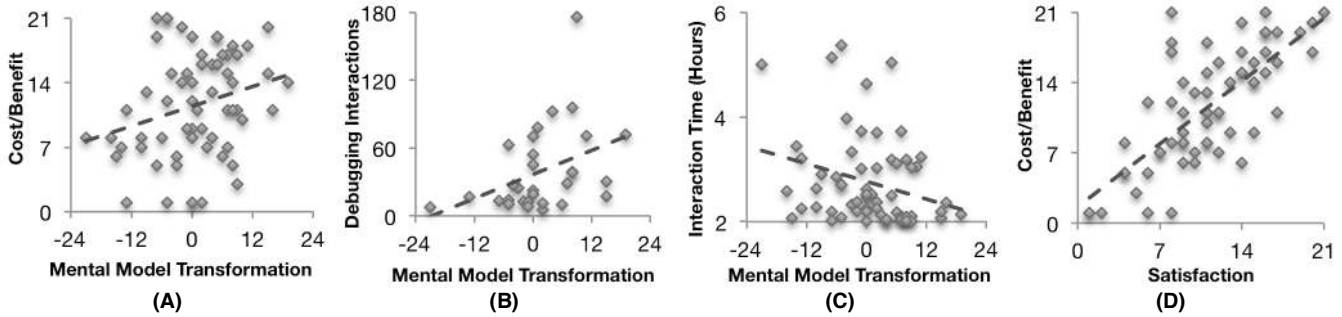

Figure 4: Scatterplots of raw data for each significant result from Table 2. Definitions for axis measurements are listed in Table 1.

Conversely, one of the participants whose mental model most increased in soundness expressed a feeling of being more in control:

"I like the idea of having more control to shape the station. Controls made sense and were easy to use. The user has a lot of options to tune the station."

Perceived cost/benefit from debugging the recommender was also significantly correlated with participant satisfaction (Table 2, row 8 & Figure 4D)—further evidence that satisfaction was indicative of an increased ability to debug the agent's reasoning. To ensure that participant satisfaction was not simply a result of time and effort invested, we tested for a relationship between reported satisfaction and the number of debugging interactions each participant performed, but found no evidence of a correlation (Table 2, row 9).

Discussion

It should be noted that one additional factor may have affected participant satisfaction. Our music database held songs by just over 5,300 artists—pandora.com, by comparison, has over 80,000 different artists [19]. Participant satisfaction may have been confounded by the fact that some participants hoped their stations would play music that was unavailable to AuPair. As one participant commented:

"The songs played weren't what I was looking for, the selection was poor. The system itself was excellent, but I need more music."

Despite this potential factor, the confluence of several metrics (cost/benefit, debugging interactions, interaction time, and satisfaction) suggests that transformations in mental model soundness translated to an improved ability to debug the recommender's reasoning, resulting in more satisfaction with AuPair's recommendations. Because our evidence suggests mental model transformations (which occurred *during* the study) helped participants debug more efficiently and effectively, continuing to provide explanations of an intelligent agent's reasoning while end users interact with the agent may help to increase their ultimate satisfaction with the agent's decisions. Such on-line explanations, however, were not investigated by the current study; we focused our exploration on the impact of explanations prior to (rather than during) user interaction with an intelligent agent.

One potential explanation of why we found no evidence that end-of-study mental model soundness was predictive of

debugging ability could be that the information presented to the With-scaffolding tutorial participants was not helpful for debugging the recommender's reasoning. Instead, the most effective participants may have learned to debug by *using* the system. However, this alternative explanation is weakened by the fact that the prototype was not transparent about how it made its decisions; the only time when participants were presented with explanations of AuPair's reasoning occurred during the With-scaffolding tutorial.

Confidence (RQ3)

Presenting a complex system to unsuspecting users could overwhelm them. We are particularly concerned with peoples' willingness to debug intelligent agents—some people (especially those with low computer self-efficacy) may perceive a risk that their debugging is more likely to harm the agent's reasoning than to improve it. Similarly, computer anxiety (a "degree of fear and apprehension felt by individuals when they consider the utilisation, or actual use, of computer technology" [4]) is known to negatively impact how (and how well) people use technology, and is negatively correlated with computer self-efficacy [29].

As Table 3 shows, almost three-quarters of the With-scaffolding participants experienced an increase in their computer self-efficacy between Day 1 and Day 5. Without-scaffolding participants, conversely, were as likely to see their computer self-efficacy decrease as to increase. A X^2 comparison showed that With-scaffolding participants were significantly more likely than a uniform distribution (in which only half would increase their self-efficacy) to increase their computer self-efficacy (X^2=6.5333, df=1, p=.011). This suggests that exposure to the internal workings of intelligent agents may have helped to allay, rather than to increase, participants' perceived risk of making their personalized agents worse.

As further evidence that it was *understanding* how the system worked (rather than simply a byproduct of using it) that influenced participants' computer self-efficacy, participants' perceived mental model soundness was significantly correlated with their computer self-efficacy at the end of the study (Pearson correlation, p<.001, r=.44, t=3.81, df=60). Additionally, there was no evidence of a correlation between the number of debugging interactions participants made and their self-efficacy at the end of the study (Pearson correlation, p=.286, r=.13, t=1.07, df=60); participants did not appear to grow more confident by simply interacting with the system. Thus, participants who at least *thought* they understood the nuances of AuPair's reasoning scored higher on the computer self-efficacy questionnaire than those who expressed little confidence in their knowledge of the recommender's logic.

	Self-Efficacy		
	Did Improve	Did Not Improve	Average Change
Without-scaffolding	16	16	3.29%
With-scaffolding	22	8	5.90%

Table 3. Participants in the With-scaffolding group were likely to end the experiment with higher computer self-efficacy than when they began.

Discussion

We hope further research will shed additional light on this preliminary link between learning how an intelligent computer program reasons, and increasing levels of computer self-efficacy (and, by association, decreasing levels of computer anxiety). Challenging tasks, when successfully accomplished, have been found to have a significantly larger impact on self-efficacy than overcoming small obstacles [2]. Personalizing intelligent agents seems exactly the sort of difficult computer task that, successfully carried out, may make people say, "If I could do *that*, surely I can do *this*…", thereby reducing the obstacles of risk and anxiety toward future computer interactions.

User Experience (RQ4)

For our final research question, we looked at the potential effects of mental model soundness on perceptions of experience, such as cognitive demands and emotional responses.

Cognitive Demands

Prior work has found that explaining concrete decisions of an intelligent agent's reasoning to end users *in situ* created an increase in participants' frustration with, and mental demand of, debugging the agent (measured via the NASA-TLX questionnaire) [14]. We suspected that end users might experience similar effects when presented with prior structural knowledge. However, the With-scaffolding participants showed no significant difference to Without-scaffolding participants' TLX scores. While acquiring a sound mental model undoubtedly requires mental effort on the part of end users, we encouragingly found no evidence that this was any greater than the mental effort required to interact with an intelligent agent without a clear understanding of its underpinnings. This suggests that end users' experience with intelligent agents does not necessarily suffer when they are exposed to more knowledge of how the agent works.

Emotional Responses

We used the Microsoft Desirability Toolkit [3] to investigate participants' user experience with the AuPair music recommender. Participants were given a list of 118 adjectives and asked to underline each one they felt was applicable to their interactions with AuPair.

The Internet General Inquirer (a tool which associates participants' words with either positive or negative connotations, based on the content analysis framework proposed in [23]) revealed that With-scaffolding participants employed slightly more positive descriptions of

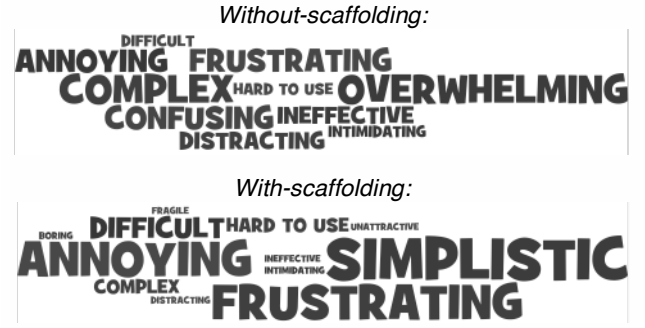

Figure 5. Tag cloud of negative descriptive terms for AuPair. Without-scaffolding participants found the system "overwhelming" and "complex" (top), whereas the With-scaffolding group (bottom) viewed it as "simplistic".

AuPair than the Without-scaffolding group (54.9% vs. 49.6%) and fewer negative descriptions (9.9% vs. 12.0%). While not statistically significant between groups, these numbers suggest that the With-scaffolding participants (with their sounder mental models) may have viewed the overall experience of interacting with AuPair in a more positive light than Without-scaffolding participants.

Participants' descriptions revealed a subtler picture of the difficulties they faced. Word clouds—in which a word's frequency is indicated by its size—of the negative descriptions show that the With-scaffolding group's complaints may have stemmed more from difficulties *using* the system than difficulties *understanding* it; these participants were apt to complain the system was "simplistic", "annoying", and "frustrating" (Figure 5, bottom), while the Without-scaffolding group appeared to have trouble even understanding the impact of their debugging interactions, citing the system as "confusing", "complex", "overwhelming", and "ineffective" (Figure 5, top).

Participants' choices of positive descriptions provide further evidence the With-scaffolding participants' mental models contributed positively to interacting with the agent (Figure 6). The phrase "easy to use" dominated their responses, alongside "innovative" and "accessible". In contrast, the Without-scaffolding participants focused on the visual appearance of the agent, with words like "clean" and "appealing". Participants with a deeper understanding of the system may have placed more emphasis on the interaction experience than aesthetics.

Discussion

Numerous benefits are associated with sound mental models, and in the case of this intelligent agent, it appears possible to gain these without impairing the user experience. This is encouraging for the feasibility of end-user debugging of recommendation systems (and possibly other types of intelligent agents), especially when the user associates a benefit with debugging the agent's reasoning.

Figure 6. Tag cloud of positive descriptive terms for AuPair. Without-scaffolding participants (top) focused on visual appearance more than With-scaffolding participants (bottom).

CONCLUSION

This paper provides the first empirical exploration of how mental models impact end users' attempts to debug an intelligent agent. By scaffolding structural models for half of our study's participants, we learned that:

- Despite the complexity inherent to intelligent agents, With-scaffolding participants quickly built sound mental models of how one such agent (a music recommender) operates "behind the scenes"—something the Without-scaffolding participants failed to accomplish over five days.
- The participants' mental model transformations—from unsound to sound—was predictive of their ultimate satisfaction with the intelligent agent's output. Participants with the largest transformations were able to efficiently adjust their recommenders' reasoning, aligning it with their own reasoning better (and faster) than other participants. These same participants were also likely to perceive a greater benefit from their debugging efforts.
- Participants presented with structural knowledge of the agent's reasoning were significantly more likely to increase their computer self-efficacy, which is known to correlate with reduced computer anxiety and increased persistence when tackling complex computer tasks.
- Participants who were presented with structural knowledge showed no evidence of feeling overwhelmed by this additional information and viewed interacting with the intelligent agent in a positive light, while participants holding only functional mental models more frequently described their debugging experience in negative terms, such as "confusing" and "complex".

This work demonstrates the value and practicality of providing end users with structural knowledge of their

intelligent agents' reasoning. Our results suggest that such an approach could better support end-user personalization of intelligent agents—telling an end user more about how it *does* work may help him or her tell the agent more about how it *should* work.

ACKNOWLEDGMENTS

We thank the study participants for their help and Weng-Keen Wong for comments on this paper. This work was supported by NSF 0803487.

REFERENCES

1. Amershi, S., Fogarty, J., Kapoor, A., and Tan, D. Examining multiple potential models in end-user interactive concept learning. In *Proc. CHI*, ACM (2010), 1357-1360.
2. Bandura, A. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review 8*, 2 (1977).
3. Benedek, J. and Miner, T. Measuring desirability: New methods for evaluating desirability in a usability lab setting. In *Proc. Usability Professionals' Association International Conference* (2002).
4. Bozionelos, N. The relationship of instrumental and expressive traits with computer anxiety. *Personality and Individual Differences 31* (2001), 955-974.
5. Compeau, D. and Higgins, C. Application of social cognitive theory to training for computer skills. *Information Systems Research*, *6,2* (1995), 118-143.
6. Echo Nest, The. http://the.echonest.com (July, 2011).
7. Fiebrink, R., Cook, P., and Trueman, D. Human model evaluation in interactive supervised learning. In *Proc. CHI*, ACM (2011), 147-156.
8. Hart, S. and Staveland, L. Development of a NASA-TLX (Task load index): Results of empirical and theoretical research, Hancock, P. and Meshkati, N. (Eds.), *Human Mental Workload* (1988), 139-183.
9. Herlocker, J., Konstan, J., Riedl, J. Explaining collaborative filtering recommendations. In *Proc. CSCW*, ACM (2000), 241-250.
10. Johnson-Laird, P.N. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Cambridge University Press (1983).
11. Kapoor, A., Lee, B., Tan, D., and Horvitz, E. Interactive optimization for steering machine classification. In *Proc. CHI*, ACM (2010), 1343-1352.
12. Kolb, D. A. *Experiential Learning*. Prentice-Hall Englewood Cliffs, NJ (1984).
13. Kulesza, T., Wong, W.-K., Stumpf, S., Perona, S., White, R., Burnett, M., Oberst, I., and Ko, A. J. Fixing the program my computer learned: barriers for end users, barriers for the machine. In *Proc. IUI*, ACM (2009), 187-196.
14. Kulesza, T., Stumpf, S., Burnett, M., Wong, W., Riche, Y., Moore, T., Oberst, I., Shinsel, A., McIntosh, K. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *Proc. VL/HCC*, IEEE (2010), 41-48.
15. Lim, B. Y., Dey, A. K., and Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proc. CHI,* ACM (2009), 2119- 2128.
16. Lim, B. Y. and Dey, A. K. Toolkit to support intelligibility in context-aware applications. In *Proc. UbiComp*, ACM (2010), 13-22.
17. McNee, S. M., Lam, S. K., Guetzlaff, C., Konstan, J. A., and Riedl, J. Confidence displays and training in recommender systems. In *Proc. INTERACT*, IFIP (2003), 176-183.
18. Norman, D. Some observations on mental models, Gentner, D. and Stevens, A. (Eds.), *Mental Models* (1983), 7-14.
19. Pandora Media, Inc. Initial Public Offering Form S-1 (2011).
20. Rosson, M. B., Carroll, J. M., and Bellamy, R. K. E. Smalltalk scaffolding: a case study of minimalist instruction. In *Proc. CHI*, ACM (1990), 423-430.
21. Sharp, H., Rogers, Y., and Preece, J. *Interaction Design: Beyond Human-Computer Interaction* (3rd edition), John Wiley (2011).
22. Sinha, R. R. and Swearingen, K. The role of transparency in recommender systems. In *Proc. CHI Extended Abstracts*, ACM (2002), 830-831.
23. Stone, P., Dunphy, D., Smith, M., Ogilvie, D., and associates. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press (1966).
24. Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. Toward harnessing user feedback for machine learning. In *Proc. IUI*, ACM (2007), 82-91.
25. Stumpf, S., Rajaram, V., Li, L., Burnett, M., Wong, W.-K., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies 67*, 8 (2009).
26. Talbot, J., Lee, B., Tan, D., and Kapoor, A. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proc. CHI*, ACM (2009), 1283-1292.
27. Tintarev, N., and Masthoff, J. Effective explanations of recommendations: User-centered design. In *Proc. Recommender Systems* (2007), 153-156.
28. Tullio, J., Dey, A.K., Chalecki, J., and Fogarty, J. How it works: A field study of non-technical users interacting with an intelligent system. In *Proc. CHI*, ACM (2007).
29. Wilfong, J. Computer anxiety and anger: The impact of computer use, computer experience, and self-efficacy beliefs. *Computers in Human Behavior 22* (2006).