An explanation system for collaborative music recommendation using graph-based visualization

Joris Schelfaut *

Faculty of Computer Science, Katholieke Universiteit Leuven, Belgium

ABSTRACT

The rationale of recommender systems is often opaque towards the end user, possibly causing decreased levels of acceptance of its recommendations. Explanation systems can overcome this problem by providing insight into the reasoning behind suggestions[7].

In this paper we will look at a white box model for collaborative filtering. This model is implemented as a visual explanation system called *SoundSuggest* which aims to explain Last.fm's collaborative recommender. The system is evaluated through a user study. We will investigate the quality of insight gaining and its effects on trust, effectiveness and persuasion of Last.fm's recommendations.

Keywords: *recommender system, insight gaining, interactive visualization, usability*

Table 1: Explanation aims. Table adapted from Tintarev and Masthoff [21].

Aim	Definition
Transparency (Tra.)	Explain how the system works.
Scrutability (Scr.)	Allow users to tell the system is wrong.
Trust	Increase users' confidence in the system.
Effectiveness (Efk.)	Help users make good decisions.
Persuasiveness (Pers.)	Convince users to try or buy.
Efficiency (Efc.)	Help users make decisions faster.
Satisfaction (Sat.)	Increase the ease of usability or enjoyment.

1 INTRODUCTION

Music catalogues for online retail have become immense over the past decades. Well-known artists and tracks make up a very small portion of this item space, which is known as the *Long-tail phenomenon*. As a result, finding new, interesting music has become a challenging task. *Recommender systems* try alleviate this problem by filtering the item repository based on a user's music taste. Taste can be modelled by analyzing user preferences and tracking user behaviour, e.g., by analyzing a user's listening history[19].

There are two commonly applied filtering strategies[16]:

- **Content-based filtering (CBF)**: Using chosen or modelled features of items to define similarity between items in the user profile and candidate suggestions;
- **Collaborative filtering (CF)**: Using overlap of item sets of each user profile to find possible suggestions in the difference of these item sets.

CF-based approaches, or hybrid approaches of CBF, CF and possibly other strategies, are often applied in music recommendation. Although these recommender systems have proven to be successful in terms of prediction accuracy, the success of recommender system also relies on the trust in its recommendations by the end user. If the user does not know why a particular item is recommended to him, the user may be reluctant to check it out. Herlocker et al. [7] describe this issue as the *black box problem*. To improve acceptance of recommendations, they propose to build an explanation system presenting the user with a *white box model* of the recommender system rationale.

This paper looks at an explanation system for collaborative music recommendation that uses a graph-based visualization. The explanation system will be evaluated based on seven aims described by Tintarev and Masthoff [21] listed in table 1. Also learnability and memorability, properties of usability as described by Nielsen[12], are also evaluated. An insight evaluation method developed by Chris North [14] is used to measure transparency. Usability evaluation methods are used to measure satisfaction, efficiency, learnability (Learn.) and memorability (Mem.). Trust, effectiveness, and persuasiveness are also evaluated during the user study. Scrutability is not supported by the explanation system.

*Joris Schelfaut - Louvain, Belgium, E-mail: joris.schelfaut@student.kuleuven.be

2 RELATED WORK AND BACKGROUND

This paper draws from concepts from the field of recommender systems, insight gaining and visualization. We will also look at the number of explanation systems that have been described in scientific literature and compare them based on the aims listed in table 1.

2.1 Collaborative filtering

Recommender system data is usually represented in the form of a matrix in which users correspond to rows, and items correspond to columns. This matrix is often referred to as the *utility matrix*. An entry $a_{i,j}$ in this matrix corresponds to a quantification of preference of user i for item j. The goal of the recommendation algorithm is to find an estimation for the blank entries in the matrix [16].

Often the utility matrix is very sparse. For systems with thousands of users and items, users will generally only have rated a small subset of those items. The problem raises significant performance issues for new users, as they have few items in their rating history, or new items, as few people have that particular item in their rating history. This problem is often referred to as the *cold start problem* [7, 16].

Another issue that is typically related to collaborative filtering, is the *gray sheep problem*. This phenomenon occurs when a user profile has no or very few other similar users associated with it. This makes it hard to establish a true 'neighbourhood' for this user[24].

2.2 Insight gaining

In [14] it is argued that insight is not a well-defined term. A formal definition might be too restrictive to capture its essence, and yet too broad to be useful. Instead, insight is considered a multidimensional property; it is complex, deep, qualitative, unexpected, and relevant[14, 23].

The quality of insight can then be determined by quantifying each of these characteristics[14]. North describes methods to evaluate insight gaining through visualizations, such as usability testing, heuristic evaluation, cognitive evaluation, and controlled experiments on benchmark tasks[14].

Chris North points out that controlled experiments suffer from problems that may hinder effective evaluation of previously listed characteristics of insight. For example the predefined nature of such experiments may decrease the amount of unexpected insight. Instead he prefers an evaluation method based on an open-ended protocol, qualitive insight analysis, and an emphasis on domain relevance[14].

Yi et al. [23] identify four processes, that are often intertwined, through which insight is established. The insight gaining processes are provide overview, adjust, detect pattern, and match mental model.

2.3 Visualization

Munzner et al. [17] identify limitations in computational and cognitive performance, and screen size for visualizing data on a screen. To alleviate these problems a wide range of visualization techniques have been developed. An overview of such techniques can be found in [10], [22], and [8].

Clutter and data overload are two problems that are common in information visualization[17]. Examples of clutter, data, and dimensionality reduction techniques are spatial distortion, clustering, change in opacity, and edge-bundling[4, 8, 9].

To describe how users interact with visualizations, Ware and Mitchell [22] list a number of *visual thinking algorithms*. A visual thinking combines perceptual and cognitive actions into a process, as the user interacts with the visualization and explores the data space[22].

2.4 Explanation systems

A number of explanation systems have been developed for recommender systems. In [15] an application called *PeerChooser* is presented by O'Donovan et al. It uses a graph-based visual explanation system for CF. Interactive elements incorporated in the visualization allow the active user to manipulate his/her neighbourhood. The *SmallWorlds* application by Gretarsson et al. [6] uses a similar approach. *Pharos* [24] also builds on ideas brought forth in [7] and [15]. The application computes a social map from the user's behaviour in content-based websites. The *TasteWeights* application by Bostandjiev et al. [1] is created for a hybrid recommendation system. It uses a graph-based approach to visualize relationships between the different recommender algorithms [1]. *SFViz* was developed by Gou et al. [5] and uses a visualization of a tag-based network to find friends based on mutual tastes in music.

Table 2 shows which of the characteristics described by Tintarev and Masthoff, were pursued for each of the explanation systems.

3 VISUALIZATION DESIGN

3.1 Translating the recommender rationale

The underlying structure of collaborative filtering, the utility matrix, can be interpreted as a *dual graph*. This is a graph $G(V, E)$ for which $V = U \cup I$ such that $U \cap I = \emptyset \wedge E \subseteq U \times I$[3]. Each non-blank entry in the utility matrix will then correspond to an edge. Figure 1 shows how a matrix is transformed into a dual graph.

Table 2: A comparison of the visual explanation systems, based on the aims by Tintarev and Masthoff listed in [21].

	Tra.	Scr.	Trust	Efk.	Pers.	Efc.	Sat.
PeerChooser	x	x		x			x
Pharos	x		x				
SFVis	x	x					
SmallWorlds	x	x		x			x
TasteWeights	x	x	x	x		x	

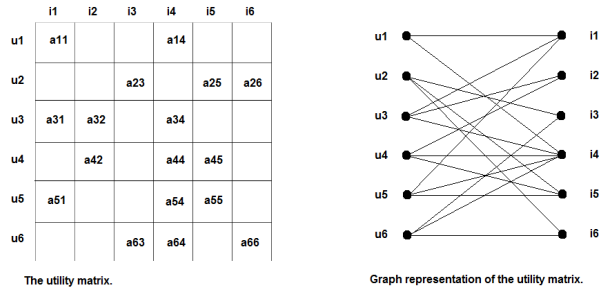


Fig. 1: Transforming the utility matrix into a dual graph: two distinct sets of nodes, users and items, only share edges between nodes of different sets.

The set of nodes U corresponds to the set of users, and the set of nodes I is set of items. In conclusion, this means that there only exist edges of that go from an item to a user or from a user to an item.

3.2 Data and dimensionality reduction

Based on a visualization design by Valdis Krebs [20], a dimensionality reduction can be performed on the dual graph through *row reduction*. One set of nodes is eliminated from the graph and is represented as implicit information in the edges. Figure 2 shows an example of this idea.

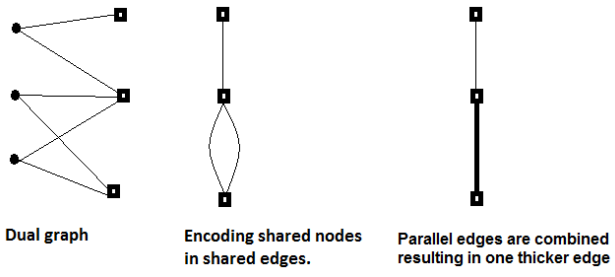


Fig. 2: A row reduction operation on each pair of edges in a dual graph will result in a dimensionality reduction where one set of nodes is removed from the graph. Additional data reduction can be achieved by clustering edges into a thicker edge. Edge thickness then depends on the number of edges involved.

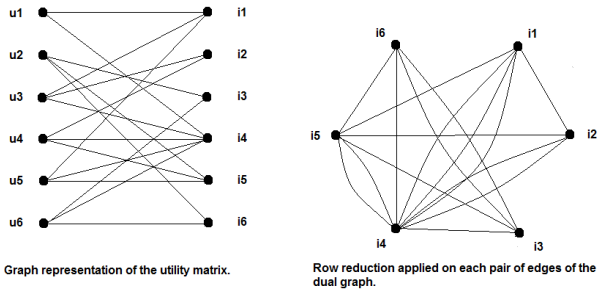


Fig. 3: Row reduction applied on the graph in figure 1.

Parallel edges are retained to keep a direct link between user and edge. In the resulting visualization of the CF-based recommender, a quantification of the similarity between users can then be established by counting parallel edges between items that occur in neighbouring profiles. Figure 3 shows how the dual graph from figure 1 is transformed into a circular graph layout with the remaining item nodes.

As it is unlikely that the whole user profile can be shown in the graph while avoiding visual clutter, the active user's favourite items are used to give a representation of the active user's profile. This way the user can still directly compare him/herself with neighbouring profiles.

3.3 Retaining contextual information

An important remark made in [11] is that data fusion algorithms can reduce information overload, but they also pose challenges to sensemaking if the human can't form an accurate mental model of the machine, to understand why and how the algorithms are doing what they are doing. Therefore, in order gain insight into the recommendation process, it is important that certain contextual information is retained. The contextual information we want to convey is two-fold:

1. The strength of the links between a recommendation and the user's profile;

2. The position of the user in his/her neighbourhood and the relation with those neighbours.

The first type of information is contained in parallel edges between items. For the second type of information, the active user's neighbours should be included in the visualization in one way or the other. In the resulting visualization shown in figure 4, the user's top neighbours are listed next to the graph. By hovering or clicking one of the listed neighbours, the relevant parts of the graph, i.e., items owned by the neighbour and the edges between them, are highlighted.

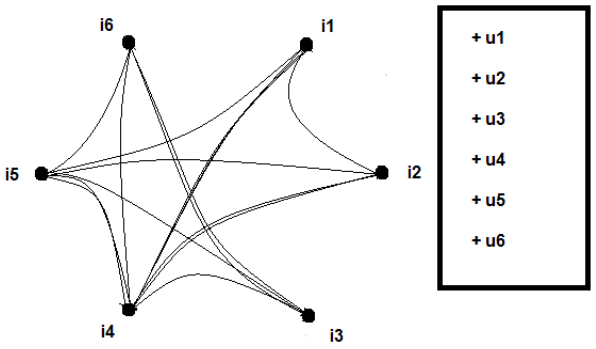


Fig. 4: The resulting visualization serving as a white box model for collaborative filtering.

4 EVALUATION

A user study was conducted to evaluate the white box model presented in section 3. In this study the explanation system aims listed in table 1 are evaluated.

4.1 Methodology

Transparency is tested by evaluating insight into the recommendation process based on North's evaluation method. We will use the think aloud protocol to obtain observational data. In particular we are looking for a user to make "domain specific inferences and hypotheses"[14].

Satisfaction, efficiency, and learnability are tested through think aloud usability testing and a summative *system usabiliy scale* (SUS) questionnaire. SUS is a *Likert scale* method consisting out of 10 questions, listed in figure 3, to investigate the subjective usability of an application[2]. Memorability is tested by asking test users that participated in previous iterations to explain the recommender rationale again at the beginning of the test.

Trust, persuasiveness, and effectiveness are evaluated through direct feedback from the test subjects.

4.2 Participants

Test users were selected from the campus and among acquaintances and were between 21 and 26 years of age. For all of the iterations combined, a total of 15 users participated in the user study of whom 12 were male and 3 were female. All users had an interest in music and listened actively to music at least once a day. Although they had a notion of what a recommender system was, none of them knew how recommendation algorithms worked.

Nielsen [13] proposes to make iterations short by limiting the number of test users. He argues that five test users per iteration are enough to find most of the usability issues. By iterating and addressing surfaced usability issues between each iteration, after a couple of iterations all usability problems will have been resolved.

Table 3: System usability scale questions.

Q1	I think that I would like to use this system frequently.
Q2	I found the system unnecessarily complex.
Q3	I thought the system was easy to use.
Q4	I think that I would need the support of a technical person to be able to use this system.
Q5	I found the various functions in this system were well integrated.
Q6	I thought there was too much inconsistency in this system.
Q7	I would imagine that most people would learn to use this system very quickly.
Q8	I found the system very cumbersome to use.
Q9	I felt very confident using the system.
Q10	I needed to learn a lot of things before I could get going with this system.

Table 4: The distribution of test users used in the evaluations for each iteration.

	Iteration			
	1	2	3	4
Number of users	5	5	5	10
From previous iterations	-	2	3	5

As a result, the test users were spread among four different iterations. Some of the users participated in multiple tests to provide direct feedback on changes made between the iterations. The distribution of users can be seen in table 4.

4.3 Prototyping

Over the four iterations the application was incrementally improved. Table 5 gives an overview of which aims were evaluated for the prototype in each iteration.

The first prototype, shown in figure 6, was made out of paper. *Paper prototypes* are relatively easy and cheap to make. Evaluation of this kind of prototype allows to detect usability problems at an early stage in the development process at a low cost. This avoids having to fix these usability problems in an actual im-

Table 5: The explanation aims that were evaluated in each iteration.

Aim	**Iteration**			
	1	2	3	4
Tra. Sat., Efc., Learn.	x	x	x	x
Mem.		x	x	x
Trust, Efk., Pers.				x

plementation, which is much more expensive[18].

The second prototype, displayed in figure 7, is an implementation of the visualization using the `D3` *JavaScript* library. This version uses the same static data as in the paper prototype. In addition to the aims listed in table 5, the success of the conversion from paper to digital prototype is evaluated as well.

The third prototype uses the visualization in a Chrome browser extension that is injected into the *Last.fm* recommendations page[1]. The visualization uses live data from the logged in user. This, in combination with the context of the Last.fm website, increases the domain relevance for insight evaluation. The main objective of the user study for this prototype was to find usability issues before moving on to a test with more users.

In the final prototype all of the other aims were evaluated, apart from scrutability, as this was not supported by Last.fm's API[2]. An example of the visualization is shown in figure 8.

4.4 Results

4.4.1Transparency

All test users were able to derive the recommendation rationale from the visualization. However, there were some differences in the speed of the insight gaining process.

The test was designed to better distinguish between the steps of the insight gaining process. To simulate the first step, *provide overview*, users were asked to form an initial mental model before interacting with the visualization. For all iterations most users saw the edges as content-based relationships, e.g. artists are connected based on genre. Only two users managed to get the visualization rationale right the first time. Based on the visualization they were able to describe the recommendation rationale.

Adjust, *detect pattern*, and *match mental model* were simulated in the next part of the test by allowing interaction with the visualization. Users could now dig deeper into the data model, gaining understanding about relationships among its elements. It should be noted that over the iterations, changes in the amount of data displayed, as well as changes in the graph's layout had an influence on the insight gaining speed.

In the first iteration, see figure 6, parallel edges were clearly visible, whereas in the other iterations, see for example figure 7, Holten's edge-bundling algorithm made parallel edges overlap. Test users indicated that this made it harder to see the link between the number of edges between an artist, and the number of highlighted neighbours. This was also supported by observational data, as this kind of stories did no longer occur in think aloud tests, unless insight into visualization's rationale had already been gained via another way.

As the number of data elements displayed increased, comparing prototypes in figures 7 and 8, it became harder to gain insight, as the total amount of edges made it harder to distinguish between them. This is probably to be expected, as small graphs are easier to analyze[8].

Usually the visualization rationale learned when the user saw that a user profile corresponded to set of interconnected nodes. By comparing overlaps between artist node sets and the highlighted active profile, they understood how neighbouring profiles were interrelated, and how recommendations were computed.

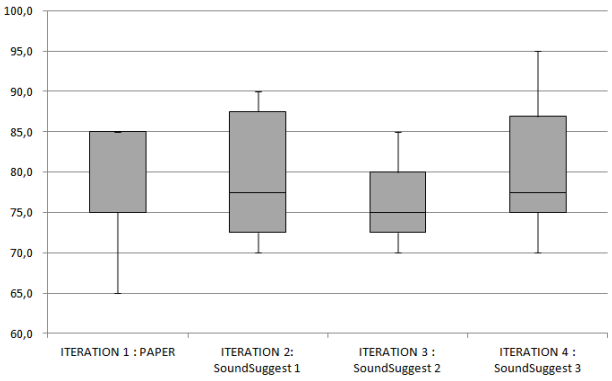


Fig. 5: The SUS results for each iteration, visualized as box plots.

4.4.2Satisfaction, efficiency, learnability and memorability

To get an idea of the perceived usability, the SUS questionnaire results are used. These changed over the iterations: new users were introduced and the design changed over time. Also, when comparing the first iteration with the others, the interaction with a paper prototype is still rather different than working with a digital version. The results are shown in figure 5.

Figures 9 to 12 show the distribution of answers for each iteration. The questions with the lowest scores overall were the first question and the seventh question.

Low scores on `Q1` may point to low satisfaction, but at the same time there were very positive remarks as well. Another explanation might be that low scores correspond to users that simply don't use Last.fm frequently. Based on the results for `Q3`, the majority of test users indicated that they thought the application was easy to use.

Although there were no negative votes for `Q7`, a lot of users were not convinced that using the application was easy to learn. Although the results for `Q10` suggest otherwise based on iteration 2 and 4, this question might be more prone to possible bias introduced dur-

[1] `http://www.last.fm/home/recs`
[2] `http://www.last.fm/api`

ing the test, which is perhaps confirmed by results for `Q4` in iteration 1 and 3.

Even though the application did not stay exactly the same over the iterations, test users that took part in two or more user tests were able to apply their experience from the previous test to the new test. Recalling the recommender rationale sometimes required some familiarization with the application again.

4.4.3Persuasion, effectiveness, and trust

In terms of trust, four scenarios were investigated: effective/bad, known/new recommendations.

It turns out the Last.fm recommender has some bias towards certain clusters of artists. Artists from a smaller music scene tend to be affected by regional effects. For example according to Last.fm, a similar artist to *De Kreuners*, a Belgian rock band, is *Samson & Gert*, a performer of music targeted to children. Another example is that bands that have certain musicians in common often are considered similar, even though their music styles are not. Usually test users already knew about these artists and indicated that this decreased their trust in the recommender system.

Similarly, bad recommendations that were new to the user also decreased trust in the recommendations.

Not all test users received this kind of biased recommendations and effective recommendations would increase the user's trust in the recommender system. Good new recommendations would increase their trust in the system more than artists they already knew about.

In some cases the explanation system helped to identify bad recommendations, as the active user's top neighbours would not have these items in their profile.

As users gained insight in the visualization and the recommender system behind it, the user's trust in the system increased as well. Persuasion was harder to measure. Typically test users would look for artist nodes where a lot of edges originated from, or users with a high similarity score.

Although the explanation system was not always as effective in helping to find good recommendations, it provided an additional means for the user to establish his/her own approach for finding recommendations. For example a user would look at neighbours for artist suggestions, rather than just the artist recommendations by Last.fm.

5 CONCLUSION AND FUTURE WORK

This paper has described a visual explanation system collaborative recommendation. The design was evaluated through user studies, first as a paper prototype which was later implemented as a Chrome Extension for the Last.fm website to explain its music recommendations. We used aims proposed by Tintarev and Masthoff, with additional usability metrics listed by Nielsen to evaluate these prototypes.

Results indicate that our design can be effective in explaining the rationale of collaborative recommendations. However, the learnability of the system still has some room for improvement. An overall SUS score in the final iteration of 80.5 suggests that the usability of the system is good, as perceived by users. Finally, the explanation system can help increase trust in the recommender system and may provide a starting point for further data exploration.

To improve the application, issues such as data density, slow data loads should be addressed further. It would also be interesting to see how the explanation system would perform for another collaborative recommender system.

REFERENCES

[1] S. Bostandjiev, J. O'Donovan, and T. Höllerer. Tasteweights: a visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 35–42, New York, NY, USA, 2012. ACM.

[2] J. Brooke. Sus - a quick and dirty usability scale. URL: `http://hell.meiert.org/core/pdf/sus.pdf`, 1996. [Online; accessed 20-March-2013].

[3] K. Dekimpe and B. Demoen. Fundamenten voor de informatica. URL: `http://people.cs.kuleuven.be/~bart.demoen/FVI/fundamenten.pdf`, 2007. [Online; accessed 9-February-2013].

[4] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, Nov. 2007.

[5] L. Gou, F. You, J. Guo, L. Wu, and X. L. Zhang. Sfviz: interest-based friends exploration and recommendation in social networks. In *Proceedings of the 2011 Visual Information Communication - International Symposium*, VINCI '11, pages 15:1–15:10, New York, NY, USA, 2011. ACM.

[6] B. Gretarsson, S. Bost, C. Hall, and T. Höllerer. Smallworlds: Visualizing social recommendations. *Eurographics/ IEEE-VGTC Symposium on Visualization 2010*, 2010.

[7] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, pages 241–250, New York, NY, USA, 2000. ACM.

[8] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, Jan. 2000.

[9] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sept. 2006.

[10] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.

[11] G. Klein, B. Moon, and R. R. Hoffman. Making sense of sensemaking 1: Alternative perspectives. *IEEE Intelligent Systems*, 21(4):70–73, July 2006.

[12] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[13] J. Nielsen. Why you only need to test with 5 users. URL: `http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/`, 2012. [Online; accessed 20-February-2013].

[14] C. North. Toward measuring visualization insight. *IEEE Comput. Graph. Appl.*, 26(3):6–9, May 2006.

[15] J. O'Donovan, B. Smyth, B. Gretarsson, S. Bostandjiev, and T. Höllerer. Peerchooser: visual interactive recommendation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1085–1088, New York, NY, USA, 2008. ACM.

[16] A. Rajaraman, J. Leskovec, and J. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2012.

[17] P. Shirley and S. Marschner. *Fundamentals of Computer Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 3rd edition, 2009.

[18] C. Snyder. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces (Interactive Technologies)*. Morgan Kaufmann, 1st edition, 2003.

[19] Y. Song, S. Dixon, and M. Pearce. "a survey of music recommendation systems and future perspectives". URL: `http://www.eecs.qmul.ac.uk/~yadings/papers/song2012a.pdf`, 2012. [Online; accessed 02-June-2013].

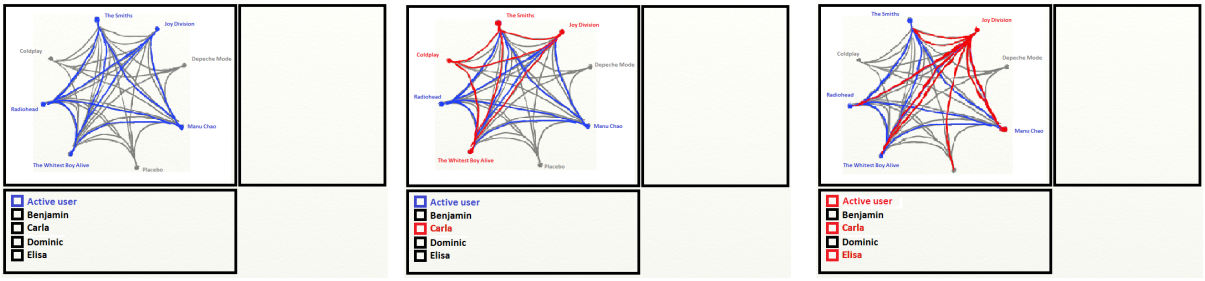
[20] J. Steele and N. Iliinsky. *Beautiful Visualization: Looking at Data through the Eyes of Experts*. O'Reilly Media, Inc., 1st edition, 2010.

[21] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, pages 801–810, Washington, DC, USA, 2007. IEEE Computer Society.

[22] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[23] J. S. Yi, Y.-a. Kang, J. T. Stasko, and J. A. Jacko. Understanding and characterizing insights: how do people gain insights using information visualization? In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, BELIV '08, pages 4:1–4:6, New York, NY, USA, 2008. ACM.

[24] S. Zhao, M. X. Zhou, Q. Yuan, X. Zhang, W. Zheng, and R. Fu. Who is talking about what: social map-based recommendation for content-centric social websites. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 143–150, New York, NY, USA, 2010. ACM.

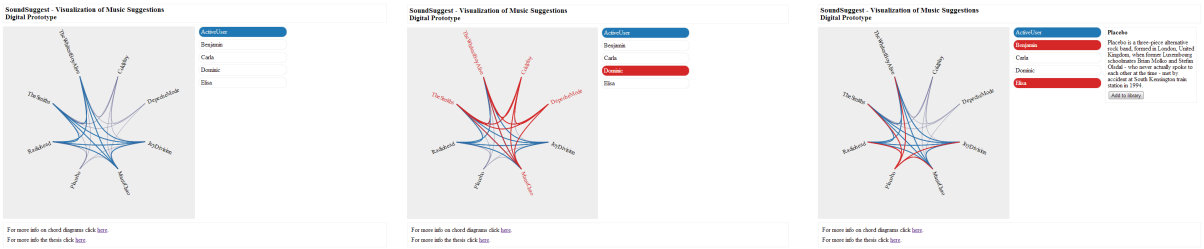


(a) The default visualization with the active user's items highlighted.

(b) Clicking a user list element.

(c) Clicking an item node.

Fig. 6: A selection of the screens used in the user study with paper prototype.

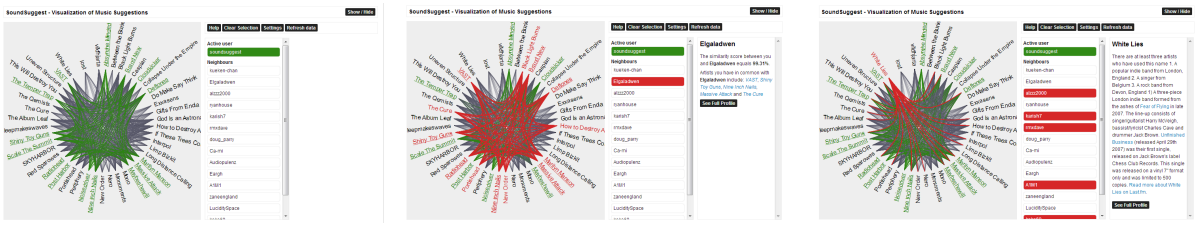


(a) The default visualization with the active user's items highlighted.

(b) Clicking a user list element.

(c) Clicking an item node.

Fig. 7: A selection of the screens used in the user study with the first digital prototype.



(a) The default visualization with the active user's items highlighted.

(b) Clicking a user list element.

(c) Clicking an item node.

Fig. 8: A selection of the screens used in the user study with the third digital prototype.

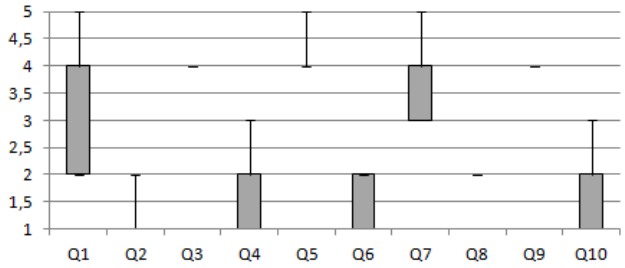


Fig. 9: The SUS results for each question for iteration 1.

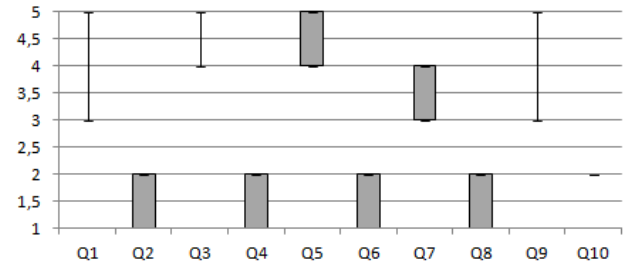


Fig. 10: The SUS results for each question for iteration 2.

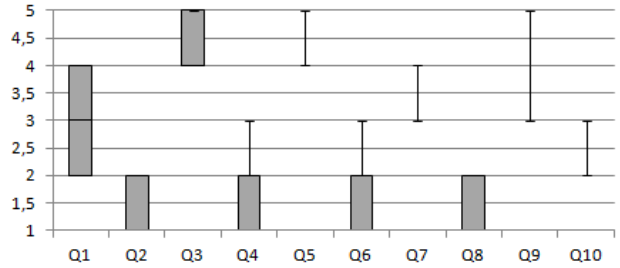


Fig. 11: The SUS results for each question for iteration 3.

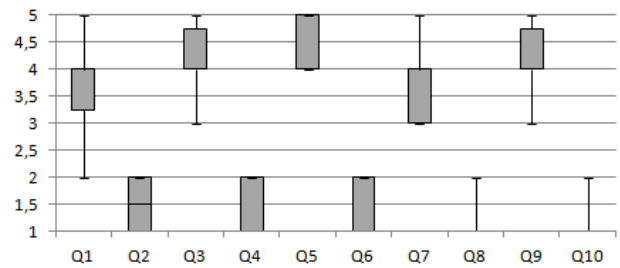


Fig. 12: The SUS results for each question for iteration 4.