IntrospectiveViews: An Interface for Scrutinizing Semantic User Models

Fedor Bakalov[1], Birgitta König-Ries[1], Andreas Nauerz[2], and Martin Welsch[2]

[1] Friedrich Schiller University of Jena
{fedor.bakalov | birgitta.koenig-ries}@uni-jena.de
[2] IBM Deutschland Research and Development GmbH
{andreas.nauerz | martin.welsch}@de.ibm.com

Abstract. User models are a key component for user-adaptive systems. They represent information about users such as interests, expertise, goals, traits, etc. This information is used to achieve various adaptation effects, e.g., recommending relevant documents or products. To ensure acceptance by users, these models need to be scrutable, i.e., users must be able to view and alter them to understand and if necessary correct the assumptions the system makes about the user. However, in most existing systems, this goal is not met. In this paper, we introduce IntrospectiveViews, an interface that enables the user to view and edit her user model. Furthermore, we present the results of a formative evaluation that show the importance users give in general to different aspects of scrutable user models and also substantiate our claim that IntrospectiveViews is an appropriate realization of an interface to such models.

1 Introduction

Adaptive Web systems are the systems that tailor their content, appearance, and behavior to the needs of individual users or groups of users. Such systems are being developed as an answer to the overwhelming and steadily growing amount of information available in typical Web systems. An adaptive Web portal, e.g., could be a portal that places on its front page links to the resources that are relevant to a user based on her interests, expertise and/or current context. The basis for such adaptation effects is a user model containing information about users, such as their interests, expertise, goals, traits, etc. [1]. In many adaptive systems the user model is considered as purely internal system information, hence it is partially or completely hidden from the user. This results in a number of grave usability problems and may well result in the user not accepting the system. For instance, it violates two of Nielsen's ten usability principles [2]. Hiding user models occludes the *system status* and hinders *control* on the adaptation, which might lead to errors, e.g. issuing irrelevant recommendations.

In order to avoid the above mentioned problems, user models need to be scrutable. This means, the user needs to be able to view and adapt the information contained in her user model [3]. Jameson [4] argued that allowing inspection and parametrization of user models are important measures to achieve predictability, transparency, and controllability of an adaptive system. According to Cook and Kay [5], the user needs to be able to understand the provenance of information in her user model, e.g., the user needs to understand why the system believes she is interested in a certain topic. Finally, Orwant [6] argued that scrutability is an essential step towards establishing trust between the user and an adaptive system. Section 2 provides a short overview of the previous research related to visualization of scrutable user models.

In Sect. 3 we introduce IntrospectiveViews, an interface that visualizes content of user models in a comprehensible way and allows users to inspect and alter them. Through this interface users can see what the system "knows" about them and how this information is used for the adaptation. Moreover, through the interface users can edit that information and control how it is used, hence achieve better adaptation effects and better control on their privacy. An important feature of IntrospectiveViews is its capability of visualizing and managing *semantic* user models. In a semantic user model, information about users is augmented with machine-understandable semantics defined, e.g., in an ontology-based domain model. Such models are more powerful than simpler user models since the system can use the semantics for interest and knowledge propagation, hence increase correctness and completeness of the user model. On the downside, these mechanisms may make adaptation decisions more difficult to understand and result in an even greater need for scrutability. In this paper, we demonstrate application of IntrospectiveViews on the example of semantic user *interest* models as presented in [7]. However, the interface can be also used for visualizing and scrutinizing other features, such as knowledge and goals.

In Sect. 4 we elaborate on a formative evaluation consisting of two parts. In the first part, we determined how important users deem visualization and editing features of a scrutable user model. These results are independent of the concrete implementation, namely IntrospectiveViews, that we evaluated in the second part of the study. The evaluation of the implementation showed that IntrospectiveViews meets the user requirements identified earlier to a large degree and is therefore a very suitable approach to our problem. Finally, Sect. 5 concludes the paper and outlines the directions for our future work.

The main contributions of this paper are twofold. First, it offers insights into user requirements towards scrutable user models. These insights are of use to anyone interested in developing such models. Second, it introduces IntrospectiveViews as an appropriate implementation of such models.

2 Related Work

A number of approaches have been proposed to visualizing scrutable user models. PeerGlass architecture [8] provides a visual method to exploring user models through a Rolodex of model planes, where each plane represents a certain type of user interests, including manually entered interests and automatically induced interests. The um_view interface [5] allows traversing through a user model by expanding the tree of leaves and viewing detailed information about the items in the model. VlUM [9] and its successor SIV [10] are capable of visualizing large user models and enable users to get an overview of the whole model, view a subset of related beliefs, filter items by relevance, and obtain detailed information about the displayed items. STyLE-OLM [11] and Flexi-OLM [12] visualize open learner models using concept graphs and trees.

One of the main distinctive features the interface described in this paper is that it *fully supports all seven tasks for information visualization* postulated by Shneiderman [13], while providing *intuitive and easy-to-use mechanisms for editing* semantic user models. In IntrospectiveViews the user can gain an overview of the entire user model and zoom into a certain part of the model to get a better view on it. It enables the user to filter out unwanted items in order to focus on the relevant ones. Additionally, it provides detailed information about a selected item when needed as well as reveals relationships among the items with respect to a number of attributes. The interface is capable of visualizing the history of changes in the model and allows extracting a certain portion of the model and saving it, so that it can be reused in other systems. Finally, the interface allows the user changing, adding, and deleting items in her user model.

3 IntrospectiveViews for User Interest Model

IntrospectiveViews is a visualization of overlay user models [1] representing user knowledge or interests. To show the features of this visualization, however, we need a concrete example. In this paper, we use the MINERVA User Interest Model developed in our earlier work [7] as this example. We will give a brief introduction to this model before delving into the details of IntrospectiveViews.

3.1 User Model and Modeling Approach

The MINERVA user interest model is implemented as an overlay model: User interests are represented as an overlay of vocabulary defined in the domain knowledge model. User interests are modeled with a hybrid approach that combines an unobtrusive method to capturing interests based on the user browsing history and a manual method allowing the user to specify her interests herself. This section provides a short description of the user interest model and the modeling approach.

We define user interest as a fact indicating that a given user is interested to a certain degree in a certain term. Here, the term is a reference to an ontology instance (e.g., company, location, or person) defined in the *domain knowledge model*, an OWL-based ontology providing machine-understandable semantics of the contained entities. The degree of interest denotes the extent to which the user is interested in a given term. We distinguish three levels of interest: *interested*, *partially interested*, and *not interested*. Also, we model user interests as time dependent features. We assume that a user might be interested in a certain term only for a certain period of time. Thus, the user interest model is represented as a collection of tuples *(U, T, I, V)*, where U is the user ID, T is the URI of an instance from the domain model, I is the linguistic variable indicating the degree of interest, V is the time period of the interest validity.

Our approach to identifying user interests involves the following processes. First, the terms indicating user interests are collected into the user model either by analyzing the content of visited pages or explicitly entered by users through IntrospectiveViews. Second, the collected terms are semantically enriched by referring them to the corresponding instances in the domain model. Finally, interest degree is determined for every term either by leveraging the term frequency, or semantic relations among the terms, or specified by the user explicitly through the interface.

3.2 IntrospectiveViews

IntrospectiveViews follows Shneiderman's Visual Information Seeking Mantra [13]: "overview first, zoom and filter, then details-on-demand". It offers users an overview over all terms present in their interest model, it allows for zooming into different parts of the model, filtering terms according to different criteria, and it will provide details, for instance additional information on a term or information about how the system determined the user interest in a term, on demand. IntrospectiveViews also supports the further tasks identified by Shneiderman: It allows revealing relationships among the items, supports browsing through the history of the user model, and allows exporting the entire model or parts of it. Let us take a closer look how this is achieved.

IntrospectiveViews, shown in Fig. 1, is implemented as a Java Applet. The interface displays user interests as term labels on a circular surface consisting of a number of colored rings. Positioning of terms on the surface, namely distance from the circle center, is determined by the terms' exact degree of interest. Here it means that the closer a term appears to the center, the higher interest it represents. Font size of terms denotes the term's frequency in the user browsing log: The terms that the user encounters often will appear bigger in relation to the terms that she reads seldom about.

Each ring, distinguished by its color, represents a certain interest group. The color scheme of rings is chosen according to the hot-and-cold metaphor, where hot, represented by red color, denotes interest and cold, represented by blue color, denotes no interest. The colors between red and blue denote partial interest. The border areas of the rings are painted in a gradient color to denote the fuzziness between the groups, i.e., uncertainty of interest degree. User interests can be grouped by their type into circular sectors, which are distinguished by different shades of gray and identified by labels containing the names of types. Such grouping allows users to place together the terms that belong to the same class, e.g., people, companies, locations, and so on.

The user is enabled to zoom in/out the entire collection of interests. By zooming in the user can get a detailed view on terms in a certain area and by zooming out she can switch back to the overview of the entire user model. In a enlarged view, the user can navigate through the collection of terms by dragging the surface in a respective direction. The interface supports a number of filtering

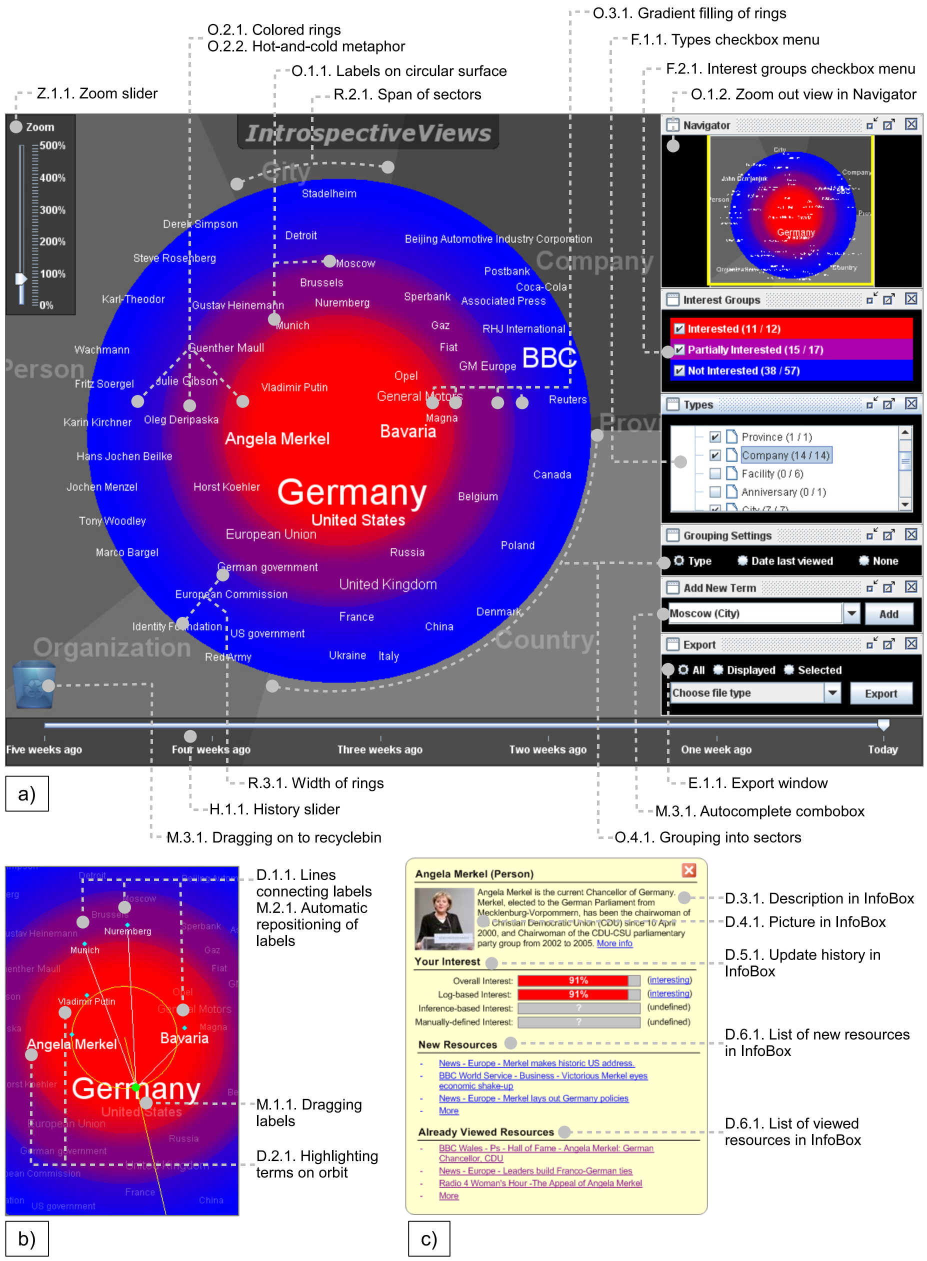

Fig. 1. Screenshots of IntrospectiveViews: (a) provides full view of the interface; (b) displays the terms that are semantically related to the selected term (white lines outgoing from Germany), the terms with the same degree of interest (highlighted on the yellow orbit), and the path along which the term can be dragged (the circle's radius represented as a yellow line); (c) displays the term's InfoBox.

Color screenshots and a screencast are available at http://www.minerva-portals.de/research/introspective-views/

options. Terms can be filtered by type and by interest group. For instance, the user can display only companies, people, and countries and/or display only the terms that she is interested in.

Also the user can obtain additional information about the terms. For instance, by pressing mouse left button on a term, the user can see its related terms (Fig. 1.b). This allows the user to find out the terms that potentially might be interesting. Right click on a term will display an infobox (Fig. 1.c) containing the term's description and picture, evidence of the user interest (whether the user has read many documents about it, or her interest was propagated from other terms, or she specified her interest manually). Also the infobox displays the list of related documents that the user has not seen yet and the list of related documents that the user has already viewed.

In addition to viewing the contents of user models, the interface enables users to edit them. In order to change interest degree in a certain term, the user can simply drag the term into an appropriate interest group (here represented by one of the rings). For instance, the user has read a number of news articles from BBC, thus this term has appeared in the user model as interesting. However, in fact the user is not really interested in BBC as a company. Using the interface, the user can simply drag the term BBC from the red ring to the blue one. This will trigger a respective update in the user model, i.e., he term will be marked as uninteresting. By dragging a term that has relevant terms, their interest will be automatically updated and their labels will be repositioned accordingly.

Also users are enabled to manually enter new terms into their user models. In the *Add new term* window, the user can simply type a term of her interest and click *Add* button. The new term will be placed in the circle (by default in the red ring). Users can remove terms by dragging them into the recycle bin. In this case, the system will stop tracking such terms and they will not appear on the circle. Respectively, this will affect the adaptation: E.g., the system will stop issuing recommendations on that term. Through the interface the user can get a retrospective view on her user model, i.e., see her interests in the past, by dragging the pointer in the *History slider*. Finally, the interface enables the user to extract a certain portion of her model and save it in a file on the disk.

4 Evaluation and Results

To evaluate usefulness and usability of IntrospectiveViews, we conducted a formative evaluation of the interface with 26 participants, 15 of which were experts in at least one of three areas of interest (user modeling and adaptation, information visualization, and usability), whereas the other 11 were non-expert users.

The aims of the evaluation were twofold. First, when designing IntrospectiveViews, we followed Shneiderman's recommendations for information visualisation [13]. Our aim was to determine how much importance users actually put on certain features of a scrutable user model: Would they like an overview? How important would it be to filter terms? Do they want to export the model? The findings of this first part of the evaluation are only very weakly coupled to

Table 1. Questionnaire (elements marked with * are implemented as a mockup)

Overview	
O.1.	The task of getting an overview of the entire user model is implemented in two ways. First, user interests are represented as labels on a circular surface (***O.1.1***), where the position denotes exact interest degree. Second, a zoomed out copy of the entire model is shown in *Navigator window* (***O.1.2.***).
O.2.	The interest groups available in the user model are represented as colored rings (***O.2.1.***), which are painted according to the hot-and-cold metaphor (***O.2.2.***).
O.3.	The fuzziness between the interest groups is represented by the gradients filling of border areas of rings (***O.3.1.***).
O.4.	To get an overview of the available types of user interests, they can be grouped into circular sectors labeled with the names of types (***O.4.1.***).
Zoom	
Z.1.	Zooming can be performed by dragging the pointer in *Zoom slider* (***Z.1.1.***) as well as by rotating the mouse wheel (***Z.1.2.***).
Filter	
F.1.	Interests can be filtered by type by selecting/deselecting corresponding checkboxes in *Display types* window (***F.1.1.***).
F.2.	Interests can be filtered by interest group by selecting/deselecting corresponding checkboxes in *Display interest groups* window (***F.2.1.***).
Details-on-demand	
D.1.	Related terms can be viewed by pressing mouse left button on a term, which will display connecting lines from the selected term to its related terms (***D.1.1.***).
D.2.	Terms with the same interest can be viewed on the orbit that is displayed when the user presses mouse left button on a term (***D.2.1.***).
D.3.	Textual description of a term can be obtained from the term's InfoBox (***D.3.1.****).
D.4.	Graphical depiction of a term can be seen in the term's InfoBox (***D.4.1.****).
D.5.	Evidence of user interest in a certain term is represented as the update history in the term's InfoBox (***D.5.1.****).
D.6.	Relevant resources that the user has not viewed yet are represented as a list of hyperlinks the term's InfoBox (***D.6.1.****).
D.7.	Relevant resources that the user has already viewed are represented as a list of hyperlinks the term's InfoBox (***D.7.1.****).
Relate	
R.1.	The relation of a term to its frequency in the browsing log is encoded in the term's font size (***R.1.1.***).
R.2.	The relation of a term type to its size, i.e., number of user interests of that type, is encoded in the sector's span (***R.2.1.***).
R.3.	The relation of an interest group to its importance is encoded in the width of the ring containing the terms of that interest group (***R.3.1.***).
History	
H.1.	View on a snapshot of the user model in the past can be obtained by setting the pointer in the *History slider* to the desired date (***H.1.1.****).
Extract	
E.1.	The entire collection of user interests or a selected part of it can be exported into a file on disk by specifying the corresponding export options in the *Export* window and clicking *Export* button (***E.1.1.****).

Table 1. Questionnaire (elements marked with * are implemented as a mockup)

Modify	
M.1.	Interest in a term can be increased or decreased by dragging the term's label closer to or further from the circle center (***M.1.1.***).
M.2.	Interest in related terms is changed automatically when the user drags a label of term that has related terms (***M.2.1.***).
M.3.	New terms of interest can be entered into the user model by typing them in the autocomplete box in *Add New Term* window (***M.3.1.***).
M.4.	Terms can be blocked from being tracked and used for adaptation by dragging them on to *Recyclebin* (***M.4.1.***).
M.5.	Modify the color scheme of the rings (the task is not implemented).

IntrospectiveViews and offer a guideline for anyone developing a scrutable user model. Second, we wanted to evaluate how well IntrospectiveViews met user requirements, i.e., whether this implementation was considered successful and how we could further improve it. To achieve these goals, the participants were first introduced to the interface and the problems that it aims to solve. Then the participants were asked to test the interface on their own and evaluate it using an evaluation questionnaire. In the questionnaire, we listed 24 tasks that the interface can support (Table 1). The tasks were classified into eight categories: *overview*, *zoom*, *filter*, *details on demand*, *relate*, *history*, *extract*, and *modify*. The first seven categories belong to Shneiderman's task by data type taxonomy for information visualizations [13]. The *modify* category was introduced to cover the edit tasks specific to IntrospectiveView. 23 out of 24 tasks are functionally implemented or implemented as mockups. Table 1 describes the eight categories of tasks and their GUI implementation. For each task, the participants were asked to cast two votes. First, how important did they find the task. This importance was rated on a 1-5 scale: from 1-not at all important to 5-very important. Second, how usable is the implementation of the task in IntrospectiveViews. The usability of the GUI implementation was rated on a 1-5 scale: from 1-not at all usable to 5-very usable. Additionally, the participants could leave free-text comments either regarding the tasks, or implementation, or the interface in general.

4.1 Results

Key goals of the evaluation were to assess the usefulness of the interface by examining the importance of the tasks it can support and to evaluate the usability of the GUI elements implementing the tasks. The overall feedback regarding the importance of tasks and the usability of GUI elements was very favorable (see Fig. 2). The participants of both focus groups provided a number of valuable free-text comments. The rest of this section details these results.

Overview. All tasks in the overview category received relatively high importance. Especially interesting was that the task of getting an overview of available types in the user model received the highest importance in its category. The participants of both focus groups confirmed our hypothesis that organizing interests

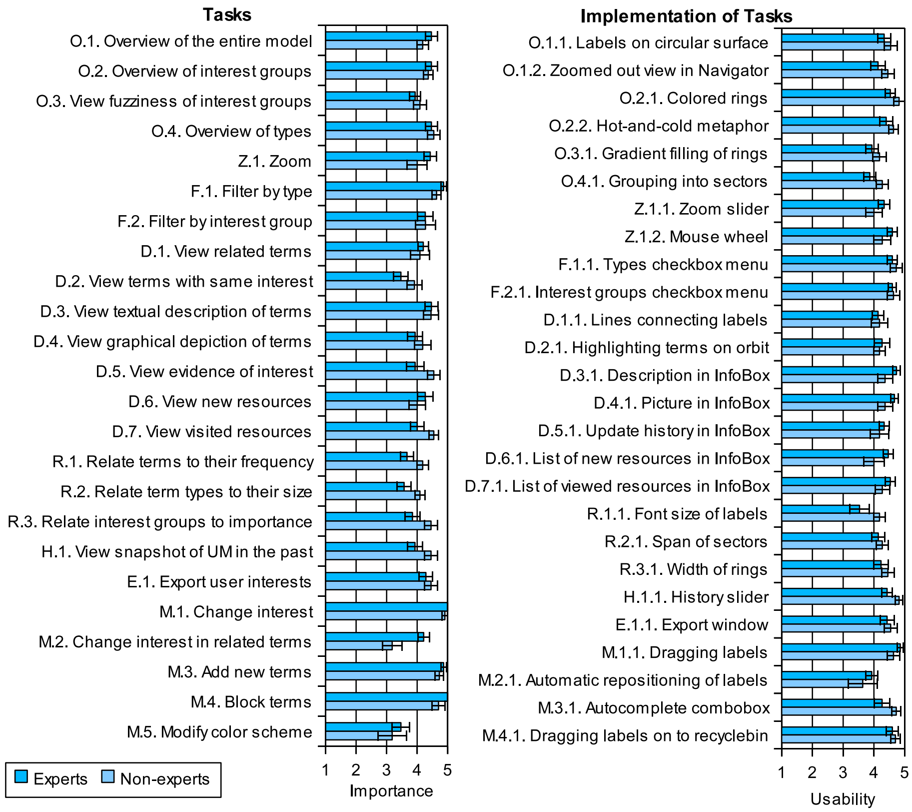

Fig. 2. Rating of importance of tasks (left) and usability of implementation (right)

by type can improve perception of the entire model. With respect to the usability of the implementation, representing user interests on a circular surface consisting of colored rings filled out according to the hot-and-cold metaphor appeared to be very intuitive and comprehensible for both experts and non-experts. However, despite the task of getting an overview of types was rated as very important, its implementation, grouping into sectors painted in different shades of grey, received the lowest usability in its category. In the free-text comments the participants of both groups mentioned that the information about types is very important, hence it must be more prominent. Experts suggested making the sectors visible not only outside of the circle, but also within it.

Zoom. The importance of zoom task was differently rated by experts and non-experts. The majority of experts considered the task as substantially important, whereas the average rating of the task's importance by non-experts appeared to be considerably lower than the experts' one. However, the usability rating of the two methods implementing the zoom task, Zoom slider and rotating the mouse wheel, was very favorable.

Filter. The average rating of both filtering tasks, filter by type and filter by interest group, is above 4, which indicates their high importance. The usability of their implementation is also rated very high. However, there was a useful suggestion from an expert on improving the usability of the filtering features. The expert suggested adding animation effects to smoothen the transition of terms when one of the filtering options is changed, e.g., smooth expansion or shrinking of sectors.

Details-on-demand. Four of the tasks in this category received relatively high importance. These are: viewing related terms, viewing textual description of terms, viewing new resources about a term, and viewing visited resources about a term. However, regarding the last two tasks, a number of respondents mentioned that their importance strongly depends on the application. For instance, one expert highlighted high importance of the task of viewing visited resources in a website that does not have a fixed navigation topology, e.g., YouTube, where it is relatively difficult to find the previously viewed content. The usability of implementation of the tasks in this category was rated good. An interesting observation is the relatively high difference between the rating of experts and non-experts with respect to the implementation of the two tasks of viewing resources about a term. Non-experts rated usability lower then the experts. In the free-text comments some of them suggested having a structure within the list, e.g., grouping by topic.

Relate. The overall importance of the relate tasks is considerably lower then the importance of tasks in other categories, in average between 3-somewhat important and 4-important. With respect to the usability of their implementation, encoding the size of a term type (number of interests of that type) into the sector's span was rated as very intuitive and usable. So was rated the encoding importance of interest groups into the ring's width. However, encoding the term's frequency into the label's font size received relatively low rating of usability. In the free-text comments, many respondents mentioned that it is unintuitive and misleading use of font size. For them, bigger font size means higher interest, which in the current implementation is not the case. It could be that uninteresting terms appear large (see example of BBC explained in Sect. 3), whereas some interesting ones can appear small.

History. The average rating of the task of viewing a snapshot of the user model in the past is above 4-important and the usability of its implementation, history slider, was also rated very high. In the free-text comments, one of the experts suggested adding a bookmarking feature to the slider, which would allow the user to bookmark certain time frames, e.g., when she was interested in Italy, and jump to this frame later on by simply clicking on the bookmark.

Extract. The importance of exporting user model was rated very high. Many respondents considered this task useful and practical if many websites support this format. The implementation, Export window, received good rating of usability.

Modify. Three modify tasks were rated as very important: change interest, add new term, and block terms. The importance of other two tasks, change interest in related terms and modify color scheme, received relatively low importance. Regarding the automatic change of interest in related terms, many respondents mentioned that the relations among the terms captured by the system do not always reflect the relations that influence their personal interest. Except for the task of changing interest in related terms, the usability of implementation of other tasks received very high rating. Most respondents rated the implementation of the interest change task, i.e., dragging labels closer to or further from the circle center, as very usable. An interesting suggestion on improving the usability of adding new terms was made by one of the experts. The expert suggested implementing this feature in a way that new terms could be added in a specific place of the model by making right click on that place and selecting terms from a popped-up menu.

General Comments. A number of participants said that it is important for them to be able to organize the terms in their user models according to their own categories. Some of them would also like to be able to define their own relations among terms. Regarding the overall usability of the interface, a number of experts requested such features as a search box for quickly jumping to a certain term and an undo feature.

5 Conclusions and Future Work

Making user models accessible to the users is a key requirement to the acceptance and success of adaptive systems. In this paper, we have presented IntrospectiveViews, a novel user interface that allows users to scrutinize user models. Following Shneiderman's recommendations, IntrospectiveViews provides users with an overview of the model as well as possibilities to zoom in, to filter, to obtain details including relationships among concepts and temporal aspects, and to export the model or part of it. Additionally, IntrospectiveViews allows the user to change the model thus actively influencing future adaptations. We have performed a formative evaluation of IntrospectiveViews using the MINERVA user interest model as an example. This evaluation showed on the one hand that the tasks supported by IntrospectiveViews are indeed deemed important by users (and should thus be supported by any scrutable user model) and showed on the other hand that the implementation of these tasks provided by IntrospectiveViews was judged as very usable for nearly all tasks. Additionally, the participants provided us with valuable feedback to further improve IntrospectiveViews.

In our future work, we will incorporate these suggestions into a new version of IntrospectiveViews and evaluate its usability with respect to completing concrete tasks in a real adaptive system. We also plan to develop implementations of other models than the one we used in this paper to further substantiate our claim that it can be widely used. Possible applications range from simply visualizing other aspects of a user model to developing visualizations comparing a person's interest to those of communities she is a member of, to using IntrospectiveViews to visualize the knowledge a student should obtain in a course and to compare it to the knowledge she has at a certain point in time.

Acknowledgements. This research is carried out in the framework of the Minerva project and is supported by IBM Deutschland Research & Development GmbH.

References

1. Brusilovsky, P., Millàn, E.: User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of LNCS. Springer, Heidelberg (2007) 3–53
2. Nielsen, J.: Enhancing the explanatory power of usability heuristics. In Adelson, B., Dumais, S., Olson, J., eds.: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, ACM (1994) 152–158
3. Kay, J.: Scrutable adaptation: Because we can and must. In Wade, V., Ashman, H., Smyth, B., eds.: Adaptive Hypermedia and Adaptive Web-Based Systems. Volume 4018 of LNCS., Heidelberg, Springer (2006) 11–19
4. Jameson, A.: Adaptive interfaces and agents. In Sears, A., Jacko, J., eds.: The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications (2nd ed.). CRC Press, Boca Raton, FL (2008) 433–58
5. Cook, R., Kay, J.: The justified user model: a viewable, explained user model. In Goodman, B., Kobsa, A., Litman, D., eds.: Proc. of the 4-th Int. Conf. on User Modeling. (1994) 145–150
6. Orwant, J.: Appraising the user of user models: Interface guidelines. In Goodman, B., Kobsa, A., Litman, D., eds.: Proc. of the 4-th Int. Conf. on User Modeling. (1994)
7. Bakalov, F., König-Ries, B., Nauerz, A., Welsch, M.: A Hybrid Approach to Identifying User Interests in Web Portals. In Erfurth, C., Eichler, G., Schau, V., eds.: Proc. of the 9th Int. Conf. on Innovative Internet Community Systems. Volume 148 of LNI., Bonn, GI (2009) 123–134
8. Kliger, J.: Model planes and totem poles: Methods for visualizing user models. Master's thesis, MIT Media Lab (1995)
9. Apted, T., Kay, J., Lum, A., Uther, J.: Visualisation of ontological inferences for user control of personal web agents. In Banissi, E., Börner, K., Chen, C., Clapworthy, G., Maple, C., Lobben, A., Moore, C., Roberts, J., Ursyn, A., Zhang, J., eds.: Proc. of the 7th Int. Conf. on Information Visualization, Washington D.C., IEEE Computer Society (2003) 306–313
10. Kay, J., Lum, A.: Building user models from observations of users accessing multimedia learning objects. In Nuernberger, A., Detyniecki, M., eds.: Adaptive Multimedia Retrieval. Springer, Heidelberg (2004) 36–57
11. Dimitrova, V.: STyLE-OLM: Interactive open learner modelling. International Journal of Artificial Intelligence in Education **17**(2) (2003) 35–78
12. Mabbott, A., Bull, S.: Alternative views on knowledge: Presentation of open learner models. In Lester, J., Vicari, M., Paraguaçu, F., eds.: Proc. of the 7th Int. Conf. on Intelligent Tutoring Systems. Volume 3220 of LNCS., Heidelberg, Springer (2004) 689–698
13. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proc. of the 1996 IEEE Symposium on Visual Languages, Washington D.C., IEEE Computer Society (1996) 336–343