User Similarity and Deviation Analysis for Adaptive Visualizations

Kawa Nazemi[1,2], Wilhelm Retz[1], Jörn Kohlhammer[1,2], and Arjan Kuijper[1,2]

[1] Fraunhofer IGD, Fraunhoferstr. 5, 64283 Darmstadt, Germany
[2] Technische Universität Darmstadt, Fraunhoferstr. 5, 64283 Darmstadt, Germany
{kawa.nazemi,wilhelm.retz,joern.kohlhammer, arjan.kuijper}@igd.fraunhofer.de

Abstract. Adaptive visualizations support users in information acquisition and exploration and therewith in human access of data. Their adaptation effect is often based on approaches that require the training by an expert. Further the effects often aims to support just the individual aptitudes. This paper introduces an approach for modeling a canonical user that makes the predefined training-files dispensable and enables an adaptation of visualizations for the majority of users. With the introduced user deviation algorithm, the behavior of individuals can be compared to the average user behavior represented in the canonical user model to identify behavioral anomalies. The further introduced similarity measurements allow to cluster similar deviated behavioral patterns as groups and provide them effective visual adaptations.

1 Introduction

The increasing amount of data in data bases and on Web poses a great challenge for the human access to relevant information. Various disciplines face the problem of human information access with different and complementary approaches. To face this problem, the area of user-adaptive visualization proposes different approaches that adapt information visualization to users' behavior. Most of these adaptive visualization approaches adapt the visual interface based on individual's user models, whereas clustering, grouping and identifying usage anomalies are commonly not investigated.

This paper proposes an approach for measuring usage similarities and deviations based on a canonical user model. We will first introduce the state-of-the-art in adaptive visualizations to outline the gap that will be filled by our approach. Thereafter we define the basic element of our approach, the canonical user model. This user model is the baseline for measuring the distance between common users and anomalies. Further it provides the ability to measure the similarity of users and create user groups. Therewith we introduce a conceptual model that measures both, deviations between the canonical user and individual users and similarities between users for user-grouping.

The main contribution of our paper is the generation of a canonical user model from implicit user interactions with visual environments to inference relevant visualization types for common users. Further these common users are applied

S. Yamamoto (Ed.): HIMI 2014, Part I, LNCS 8521, pp. 64–75, 2014.

as baseline for identifying groups and in particular deviations. Such deviations can be used to infer anomalies in usage behavior and identify experts and novices.

2 Related Work

The term *adaptive visualization* is used for different levels of adapting the visual representation, filtering and recommending data to be visualized. Golemati et al. [1, 2] introduced a context-based adaptive visualization that concerns user profiles, system configuration and the document collection (data set) to provide an adequate visualization. They state that the choice of 'one' adequate visualization from a pool of visualizations leads to a better performance. The adaptation of the visualization is based on the "context" which has to be generated manually [1]. An implicit interaction analysis is not performed; further the use user similarity or deviations is not is not investigated. A similar approach is proposed by Gotz et al. with the HARVEST tool [3]. HARVEST makes use of three main components: a reusable set of visualization widgets, a context-driven visualization recommendation and semantic-based approach for modeling user's analytical process. Here the limitation is the need of experts who have to define an initial design for the interaction patterns and the resulting visualization recommendation [3]. With the *APT tool* [4] and the consecutive *Show Me* system [5], Mackinlay et al. differ from the previously described works in a metaphor of small multiple displays and an enhanced aspect of user experience in visual analytics. Although they propose an adaptive visual system, the used algebra is defined for data to provide a better mapping of data-tables to visual representations. Another approach for data-adaptive visual presentation is HiMap [6]. The system reduces the graph-layout complexity (visual density) by an adaptive data-loading algorithm. Similarly, da Silva et al. investigated the reduction of complexity by adapting the data [7]. The adaptation of a spatial visual presentation layer based on user preferences is proposed in the *Adaptive VIBE* system by Ahn and Brusilovsky ([8], [9]). Their approach uses algorithms for identifying data similarities [8]. The introduced examples demonstrate the upcoming popularity of adaptive visualization concepts. However, the majority of the systems require the involvement of either experts to model an initial visualization design or the active involvement of users'. The use of canonical user model with similarity and deviation analysis are not considered for training the adaptive behavior.

3 User Modeling by Interaction Analysis

The measurement of users' similarities or deviation can be performed through a consistent representation of users' behavior. Our approach targets at modeling users through their interaction behavior with a visualization. This section introduces some concepts of our interaction analysis algorithm introduced in [10] and [11] to enable a comprehensible picture of the entire user analysis procedure.

3.1 Formal Representation of Users' Interactions

The adaptation of information visualization requires the acquisition of users' informational context. To provide such a context, we introduced in [10–12] an interaction analysis algorithm that allows the analysis of users' interaction to model a behavioral pattern and provide interaction predictions. This paper will use parts of the algorithm to model users and measure similarities and deviations between users. According to [13] we first define an interaction event I as a relation instantiated with leaf values of the domains equivalent to *Relational Markov Models* as $I = r(k_1, ..., k_n), \quad k_i \in leaves(D_i) \quad with \quad 1 \leq i \leq n.$ Thereby $leaves(D_i)$ are the leaf nodes of the domain D_i and r is a relation over the domains $D_1, ..., D_n$ [10]. This formal representation of users' interaction enables to model each interaction in a unique way and analyze them to model the behavior or measure predictions and probabilities.

3.2 Deriving Users' Interaction Behavior

Users' interaction behavior is the way hoe users interact with a system to achieve theirs goals. This behavior can give us information about preferences in system use or even indicates the expertise level of users. The users' interaction behavior can be described as the probability distribution of users' interactions in contrast to the entire possible interactions with the system. To compute the probability distribution of users' interactions, we first determine the Steady State Vector (SSV) as a relative measurement for the occurrence of interaction events. The SSV is a normalized probability distribution with $\sum_{i=1}^{n} p_i = 1$ and therewith a probability distribution over the entire possible interactions. We use the frequency distribution of the interactions. The frequency distributions is computed based on the quantitative occurrence of an interaction i in contrast to the entire interactions. Therewith the probability for occurrence of an interaction i is defined as $p_i = \frac{v_i}{|A|}$, where v_i is the amount of all occurrences of the interaction i and $|A|$ is the amount of interactions the user performed previously. We use for the set of all previous interactions A either the set of interactions of the individual or canonical user [14].

The formal representation of the interactions provides context information of the interaction events. Abstractions of interaction events are defined as sets of interaction events by instantiating the relation r with the inner nodes of the domains [10, 13]. There a frequency distribution and therewith a probability of all domains can be computed on each degree of the domain abstraction [13]. Based on the defined quantitative occurrence measurement, we define the function $quant(depth_{D_1}, ..., depth_{D_k})$, where $depth_{Di}$ is the level of abstraction for every domain D_i as the hierarchical level of the domain, starting with 0 for the highest level. With each occurrence of the function $quant(depth_{D_1}, ..., depth_{D_k}))$ a set L of the abstraction levels is generated illustrated according to the abstraction levels of Anderson et al. [13, p. 3] $L = \{r(\delta_1, ..., \delta_k)\}$, with $\delta_i \in nodes_i(depth_{D_i})$ and $0 \leq depth_{D_i} \leq maxDepth(D_i)$, and $1 \leq i \leq k$.

Thereby $nodes_i(depth_{Di})$ are the nodes of the domain D_i with the depth $depth_{Di}$, and the maximum abstraction level of the domain D_i is defined as $maxDepth(D_i)$. In our case we instantiate this function with $k = 3$ thus, we have the three domains of *Device*, *Visual Layout* (SemaVis), and *Data*. Therewith the function is used as $quant(depth_{D_1}, depth_{D_2}, depth_{D_3})$. The type of users' interaction is based on the *Device* and the targets to be achieved, here a predefined taxonomy is given. The second domain of *SemaVis* is representing the different visual layouts. The visualization environment contains an enhanceable set of visual layouts with a predefined taxonomy. The third domain of *Data* contains the semantic hierarchy of the data entities. The semantic hierarchy of the data is gathered by an iterative querying approach [15] and used as taxonomy for this particular domain. With the automatic inclusion of the semantic hierarchy and the generated taxonomy on inheritance-level any changes of the database can be performed without restrictions, thus the underlying semantics provides appropriate structure for the formal representation of the user interactions.

The probability p_α for each abstraction $\alpha \in L$ is calculated with the probabilities from the SSV $\boldsymbol{s}$ [10] as $p_\alpha = \sum_{q_i \in \alpha} ssv(q_i)$, thereby $ssv(q_i)$ is the probability of the interaction q_i from the SSV. Hence the result is a probability distribution over sets of interaction events.

The probabilistic distribution of users' interactions over the different levels of abstraction enable us to measure various values for preferences and knowledge of the users. Modeling users' can be performed on a detailed level by investigating all abstraction levels of the three identified domains.

3.3 Modeling Users

The main goal of analyzing users' interaction behavior with data and visualizations is to represent these for generating an abstract model of the behavior and provide sufficient visual adaptations. Sleeman proposed that the main aspects in modeling users are the *nature* and the *structure* [16]. Thus nature refers to user characteristics or feature that are in our approach gathered just implicitly from users' interaction, we constrain these features to users' interest and tasks [16–18]. The structure of the user model should be transferable to other domains of knowledge (data-sets) and should therewith enable the use of the model in various data domains.

With the introduced SSV and the various levels of abstraction, we already defined an abstract model of the user. The SSV represents the probability distribution of users' interaction in different level of abstraction. Further it refers to three dimensions: the used device and type of interaction, the visual layout, and the data [12].

For a more comprehensible illustration for modeling users and to enable in the next steps measuring users' similarities and deviations with the canonical user model, we introduce some general definitions, that are used throughout this paper. We define the set $U = \{u_1, u_2, ..., u_n\}$, where each u is a user. Additionally, we define the set $V = \{v_1, v_2, ..., v_k\}$ with each v being a visual layout of all visual layouts from the first abstraction level of the visual layout domain D_2. Further we define d being a data element from the set of all data elements D of all abstraction levels from the data domain D_3. For considering the users' behavior on individual user level, we extend the equation described in Section 3.2 by allowing the look-up from the SSV ssv_u of each individual user $u \in U$ as follows:

$$p_{u,\alpha} = \sum_{q_i \in \alpha} ssv_u(q_i) \tag{1}$$

Furthermore we introduce $p_{u,v,d}$ as a short form to extract the probability of an individual user for the correlation of a visual layout v and a data element d.

$$p_{u,v,d} = p_{u,r(device,v,d)} \tag{2}$$

Although the interaction type *Device* D_1 is gathered in each users' interaction, we dismiss this information in this context and use the abstraction level 0. This lets us extract the relevance value of the data element d in combination with the visual layout v for a specific user u. In the next step, we introduce two relevance vectors for each user in the user model. The *visual layout usage- vector* $\boldsymbol{vl}$ contain the relevance values of visual layouts according to their usage of each user and provides us information about users' "visual layout preferences" that is again a probability distribution of the interaction behavior with the visual layouts. The *data interests- vector* $\boldsymbol{di}$ contain the relevance values of the data elements according to the interest and previous knowledge of the individual users. Each entry $p_V(u,v)$ in the $\boldsymbol{vl}$ of an individual user contains the normalized relevance values of each visual layout $v \in V$ and is calculated as follows:

$$p_V(u,v) = \frac{\sum\limits_{d \in D} p_{u,v,d}}{\sum\limits_{d \in D} \sum\limits_{v_i \in V} p_{u,v_i,d}} \tag{3}$$

The creation of $\boldsymbol{di}$ for each user uses the semantic inheritance relations in addition to the relevance values between visual layouts and data elements. Let, as previously stated, $p_{u,d,v}$ be the relevance value of an individual user $u \in U$ for a data element d in combination with a visual layout v and let $S_d \subseteq D$ be a set of all data elements, which have a semantic relation with data element d. The relevance value $p_D(u,d)$ of a individual user $u \in U$ for a data element d is calculated as follows:

$$p_D(u,d) = \max_{v \in V} p_{u,v,d} + \sum_{d_i \in S_d} \frac{\max\limits_{v \in V} p_{u,v,d_i}}{|S_{d_i}|} \tag{4}$$

These relevance values of the individual data elements form $\boldsymbol{vl}$. In contrast to $\boldsymbol{vl}$, the individual relevance values between data elements and visual layout are not added up while creating the vector. Instead, the visual layout relevance values is being used, which has the highest value for the corresponding data element.

With the introduced definition so far, the users' interest and preferences are modeled. For determining the tasks, we use as described in [10] the occurrences of similar interaction sequences O as behavioral patterns. To model a "training file" that is continuously updated by users' behavior and leads to a more efficient way of identifying the occurrence of similar and frequent sequences, we use the *canonical user model* [14]. Our algorithm make use of the interactions as relations between data and visual layout, whereas the interaction type *Device* is investigated too as the first domain of the SSV D_1. This domain can be used to determine different dependencies on transition level. The users' interactions with data and visualization may have different relevance. To gather the information the first domain $D1$ of the SSV is used that provides information about the type of users' interaction, e.g. as *Device.Mouse.selectVis*. The procedure allows to weight successful interactions that leads to achieving the goal or explicit selecting visual layout higher than those interactions that lead to removing visual layouts. This procedure is coupled to the activity or task recognition of the algorithm proposed in [10] and modeled through the general and frequent behavior of the canonical user. Therewith the formal description can be derived from the users' interaction behavior and the prediction of users' action as described in [10] and [11]. Figure 1 illustrates schematically the described procedure.

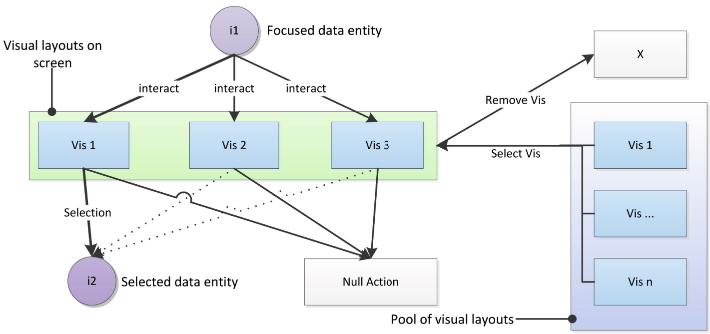

Fig. 1. Schematic illustration of the interaction relevancies

The Canonical User

The canonical user model represents the average users' behavior with the visualization system. This user model is the baseline for adapting the visual layout and data for all users and improves therewith the general usage of the visualization system. Thus it is used on the one hand for the general adaptation and on the other hand for measuring certain behavioral deviations and anomalies for individual user, it is one of the core components of our approach. Every user that interacts with the visualization environment pulls one's weight to the canonical user model. The interaction of each user, even if the adaptivity of the system is disabled, contributes to this model.

To describe the interaction behavior and therewith the probability distribution for the canonical user in context of visual layouts, we use the probability values $p_{u,v,d}$. Based on these probabilities the interaction behavior of a canonical user can be computed as illustrated in equation 5. Where the sum of interaction probabilities of each user $u \in U$ with a certain visual layout v is divided by the amount of all users $|U|$.

$$can_V(v) = \frac{1}{|U|} \sum_{u \in U} p_V(u, v) \tag{5}$$

A similar correlation can be built between users and data. The main difference is that the leaf nodes are investigated in different levels of abstraction. Thereby either a data or information entity can be a leaf node or an intermediate node of the entire taxonomic structure. We previously defined $d \in D$ as a data element from a set of all data elements in all abstraction levels from the data domain D_3. Additionally, for the canonical user, only users who interacted with the specific data are considered. Therefor we do by the amount of users, who interacted with this data entities.

$$can_D(d) = \frac{1}{|\{u | u \in U, p_D(u,d) > 0\}|} \sum_{u \in U} p_D(u, d) \tag{6}$$

User Group and Role Definition

Similar interaction behavior of users can be used to define *User Groups*. The identification of these groups leads to define certain user roles based on the interaction behavior of the users. To define user groups and roles, we use two methods: In the first method, users are clustered based on their usage of the visual layouts V, in the second method, their interest in certain data or knowledge dimensions is the basis for the clustering. A specific user is assigned to a cluster based on the following definition. Let $sim(c, u)$ be a function, which provides the similarity of a user u to a cluster c of all clusters $C = \{c_1, c_2, ..., c_n\}, c_i \subseteq U, \forall c_i, c_j : i \neq j, c_i \cap c_j = \emptyset$. Here, a higher value means stronger similarity. A user is assigned to a cluster c, if there is no other cluster $c_i \in C, c \neq c_i$, that has a stronger similarity with the users individual previous interactions: $\forall c_i \in C, c_i \neq c : sim(c_i, u) < sim(c, u) \rightarrow u \in c$. The average value of each cluster c of all visual layout cluster C_V is calculated in the same way as for the canonical user. With the main difference that only users in their respective clusters are considered. The normalized value $p_V(c, v)$ of a visual layout v of a cluster c is calculated as $p_V(c, v) = \frac{1}{|c|} \sum_{u \in c} p_V(u, v)$. The average value of each individual cluster c of all data domain clusters C_D is also calculated similar to the calculation of the canonical user. Additionally, the normalization only considers users, who contributed to the calculated value. The measurement of this normalized average value $p_D(c, d)$ of a data entity d of a cluster c is calculated as $p_D(c, d) = \frac{1}{|\{u | u \in c, p_D(u,d) > 0\}|} \sum_{u \in c} p_D(u, d)$.

Figure 2 illustrates the previously presented abstract depiction of the user model with additional clustering. The "+"-signs represent the center of each cluster. The individual clusters are aligned in the form of rays radial around the canonical user in the center of the figure. With this procedure different user clusters can be determined automatically, even if the clusters are not labeled. Figure 3 illustrates the user grouping in a more hierarchical way to outline the relationship to the canonical user model. Every user, regardless, if he or she belongs to a group, provides interaction information to the canonical user model and their models inherit from the canonical user model. Grouped users inherits further from the average group user model and provides interaction information to the average group user models too.

4 User Similarity Analysis

User similarity measurements allow the comparison of individual users through a numerical quantity value [19, 20]. This value can be used to measure how similar users are according to their behavior. These measurements are used for the calculation of the similarity between a user and a user group in addition to the similarity of two different users. Figure 2 illustrates abstractly the similarity between users and user groups. The angle between two objects represents their similarity to each other. A smaller angle means more similar objects. The green gradient illustrates regions of high to low similarity of the selected user (blue icon) to other users in these regions.

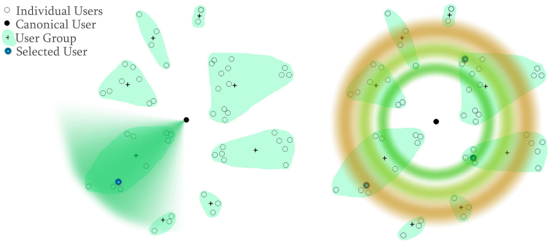

Fig. 2. User similarity and deviation analysis based on the canonical user

The basis for the calculation of the similarity are the two previously described vectors ($\boldsymbol{vl}$ and $\boldsymbol{di}$). Their composition differ, $\boldsymbol{vl}$ is normalized and not very sparse after a short usage of the visual system by a specific user, because commonly new visual layouts are only added in large time intervals. In contrast to that, $\boldsymbol{di}$ is very sparse, even after a very thorough usage of the visual system by the specific user. Because many and new data elements can be added or removed continuously. This is the main reasons, why different similarity measurements are used to determine the similarity between users and other users or user groups.

For the calculation of the similarity between users on the basis of their data interests relevance values, the *Pearson Correlation Similarity* [19–21] metric is used. Let $u_a \in U$ and $u_b \in U$ be two users and $p_D(u_a, d)$, $p_D(u_b, d)$ their respective relevance values for the data element $d \in D$. The similarity between these two users $sim_D(u_a, u_b)$ is calculated as follows:

$$sim_D(u_a,u_b)=\frac{\sum\limits_{d\in D_{ab}}(p_D(u_a,d)-\overline{p_D(u_a)})(p_D(u_b,d)-\overline{p_D(u_b)})}{\sqrt{\sum\limits_{d\in D_{ab}}(p_D(u_a,d)-\overline{p_D(u_a)})^2}\sqrt{\sum\limits_{d\in D_{ab}}(p_D(u_b,d)-\overline{p_D(u_b)})^2}}\mid D_{ab}=D_{u_a}\cap D_{u_b} \tag{7}$$

Here, $\overline{p_D(u)} = \frac{1}{|D|}\sum\limits_{d\in D} p_D(u,d)$ is the mean value of all values in $\boldsymbol{di}$ for a user $u \in U$.

The calculation of the similarity between users on the basis of their visual layout relevance values also uses the *Pearson Correlation Similarity* metric. But here, no normalization with the mean value of the respective vector occurs, because these vectors are already normalized. The value for the similarity of two users $u_a \in U$ and $u_b \in U$ and their respective relevance values $p_V(u_a, v)$ and $p_V(u_b, v)$ for the visual layout v is calculated as illustrated in Equation 8.

$$sim_V(u_a,u_b)=\frac{\sum\limits_{v\in V_{ab}} p_V(u_a,v)p_V(u_b,v)}{\sqrt{\sum\limits_{v\in V_{ab}} p_V(u_a,v)^2}\sqrt{\sum\limits_{v\in V_{ab}} p_V(u_b,v)^2}}\mid V_{ab}=V_{u_a}\cap V_{u_b} \tag{8}$$

The choice of the *Pearson Correlation Similarity* metric for the determination of the similarity between users and other users or user groups is based on the fact, that the *Pearson Correlation Similarity* only considers elements, which have relevance values on both sides.

5 User Deviation Analysis

The user deviation represents the difference in user behavior of each individual user to the average behavior of the canonical user. It is assumed, that users can also be similar to each other, if they differ similarly in their interaction behavior from the average behavior. Thereby their direct similarity to each other is not measurable. This can happen, if e.g. the adaptive system could not yet determine the overlapping interests for the particular users.

Figure 2 illustrates an abstract depiction of the deviation of users' behavior from the average behavior of the canonical user in addition to the previously calculated user groups from the similarity analysis. The distance of the user to the canonical user and accordingly the radius represent the aforementioned behavioral deviation. For a selected user, the gradient of the ring symbolizes the region with very similar deviation in behavior.

The calculation of this behavioral deviation is also based on the two previously described vectors ($\boldsymbol{vl}$ and $\boldsymbol{di}$), which were also used for the modeling of the canonical user and for the creation of the user groups.

Unlike the calculation of the similarity between the users, we used the *Cosine Similarity* metric [20–23] for calculating the behavioral deviation. This is because the consideration of the relevance values that are not common for both users are relevant for the measurement of the deviation. Since the *Cosine Similarity* metric does not perform a normalization, the calculation of $\boldsymbol{vl}$ and $\boldsymbol{di}$ are identical.

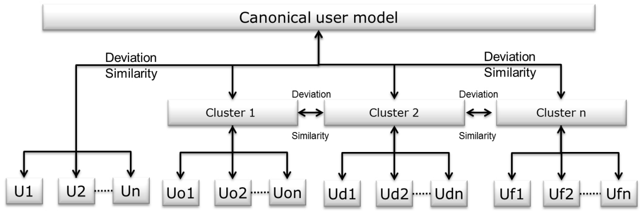

Fig. 3. Deviation and similarity relations to the canonical user model

Let $p_D(u, d)$ be the relevance value of a data element d of the data element set D for a user $u \in U$ and $can_D(d)$ the relevance value of the canonical user for the data element d. The information deviation (interest-deviation) $dev_D(u)$ of a user u can be calculated as follows:

$$dev_D(u) = \frac{\sum_{d \in D} p_D(u,d) can_D(d)}{\sqrt{\sum_{d \in D} p_D(u,d)^2} \sqrt{\sum_{d \in D} can_D(d)^2}} \tag{9}$$

This leads to the definition of a similarity between two users based on their interest-deviation from the canonical user $sim\_dev_D(u_a, u_b)$, which can be expressed as follows:

$$sim\_dev_D(u_a, u_b) = 1 - |dev_D(u_a) - dev_D(u_b)| \tag{10}$$

Equivalently, the similarity between two users based on the deviation in the usage of visual layouts to the canonical user can be calculated as follows:

$$dev_V(u) = \frac{\sum_{v \in V} p_V(u,v) can_V(v)}{\sqrt{\sum_{v \in V} p_V(u,v)^2} \sqrt{\sum_{v \in V} can_V(v)^2}} \tag{11}$$

$$sim\_dev_V(u_a, u_b) = 1 - |dev_V(u_a) - dev_V(u_b)| \tag{12}$$

The returned value is in the range between one (identical distance) and zero (completely different distance). The analysis of deviations is performed between the canonical user and either one individual user or the average user of a certain identified group.

6 Conclusion

This paper introduced an approach for modeling the average users' behavior within visual environments to improve the general adaptation effects. Therefore, the canonical user model and its formal representation was introduced. Based on this the deviance measurement was illustrated. We showed that the measured distance between the canonical user and individual users can be used to detect anomalies in user behavior, which may lead to "screw down" the adaptation effect. If the distance between the user behavior is similar to that of the canonical user, the canonical user model can be taken as user model for new situations, visual layouts, users, or data-bases. Further the distance can be used to determine the application of the canonical user to new situations, visual layouts, or data bases for user groups. These groups are determined by applying similarity algorithms on users' behavior and provide an "average user model" that can be used for measuring the distance between the group and the canonical user. Based on this measurement further aspects can be determined: In this case of similar user behaviors the average user model of the group inherits the canonical behavior or in case of large distance the adaptation effects are reduced until enough information of the users' in that particular group are gathered. Further the measured similarities between same distanced users (Figure 2) are used to detect similarities as described in the Section 4. If the distance of two users or average user groups are similar according to the canonical user model, the similarity algorithm is applied to measure certain similarities between the groups or users. If similar behavioral patterns are detected and one of the user groups contains certain information that is missing in the other group, these information are applied to extend the average user model of that group (Figure 3).

Acknowledgments. This work has been carried in the FUPOL project, partially funded by the European Commission under the grant agreement no. 287119 of the 7th Framework Programme. This work is part of the SemaVis technology, developed by the Fraunhofer IGD (`http://www.semavis.net`).

References

1. Golemati, M., Halatsis, C., Vassilakis, C., Katifori, A., Lepouras, G.: A context-based adaptive visualization environment. In: Proceedings of the Conference on Information Visualization, IV 2006, pp. 62–67. IEEE Computer Society, Washington, DC (2006)
2. Golemati, M., Vassilakis, C., Katifori, A., Lepouras, G., Halatsis, C.: Context and adaptivity-driven visualization method selection. In: Mourlas, C., Germanakos, P. (eds.) Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies, pp. 188–204. IGI Global (2009)
3. Gotz, D., When, Z., Lu, J., Kissa, P., Cao, N., Qian, W.H., Liu, S.X., Zhou, M.X.: Harvest: An intelligent visual analytic tool for the masses. In: Proceedings of the First International Workshop on Intelligent Visual Interfaces for Text Analysis, IVITA 2010, pp. 1–4. ACM, New York (2010)
4. Mackinlay, J.: Automating the design of graphical presentations of relational information. ACM Trans. Graph. 5, 110–141 (1986)
5. Mackinlay, J., Hanrahan, P., Stolte, C.: Show me: Automatic presentation for visual analysis. IEEE Transactions on Visualization and Computer Graphics 13, 1137–1144 (2007)
6. Shi, L., Cao, N., Liu, S., Qian, W., Tan, L., Wang, G., Sun, J., Lin, C.Y.: Himap: Adaptive visualization of large-scale online social networks. In: Visualization Symposium, PacificVis 2009, pp. 41–48. IEEE Pacific (2009)
7. da Silva, I., Santucci, G., del Sasso Freitas, C.: Ontology Visualization: One Size Does Not Fit All. In: EuroVA 2012: International Workshop on Visual Analytics, pp. 91–95. Eurographics Association (2012)

8. Ahn, J.W., Brusilovsky, P.: Adaptive visualization of search results: Bringing user models to visual analytics. Information Visualization 8, 180–196 (2009)
9. Ahn, J.W.: Adaptive Visualization for Focused Personalized Information Retrieval. PhD thesis, School of Information Sciences, University of Pittsburgh (2010)
10. Nazemi, K., Stab, C., Fellner, D.W.: Interaction analysis for adaptive user interfaces. In: Huang, D.-S., Zhao, Z., Bevilacqua, V., Figueroa, J.C. (eds.) ICIC 2010. LNCS, vol. 6215, pp. 362–371. Springer, Heidelberg (2010)
11. Nazemi, K., Stab, C., Fellner, D.W.: Interaction analysis: An algorithm for interaction prediction and activity recognition in adaptive systems. In: Proc. of IEEE ICIS, pp. 607–612. IEEE Press, New York (2010)
12. Nazemi, K., Stab, C., Kuijper, A.: A reference model for adaptive visualization systems. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part I, HCII 2011. LNCS, vol. 6761, pp. 480–489. Springer, Heidelberg (2011)
13. Anderson, C.R., Domingos, P., Weld, D.S.: Relational markov models and their application to adaptive web navigation. In: KDD 2002: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 143–152. ACM, New York (2002)
14. Nazemi, K., Retz, R., Bernard, J., Kohlhammer, J., Fellner, D.: Adaptive semantic visualization for bibliographic entries. In: Bebis, G., et al. (eds.) ISVC 2013, Part II. LNCS, vol. 8034, pp. 13–24. Springer, Heidelberg (2013)
15. Nazemi, K., Breyer, M., Forster, J., Burkhardt, D., Kuijper, A.: Interacting with semantics: A user-centered visualization adaptation based on semantics data. In: Smith, M.J., Salvendy, G. (eds.) Human Interface, HCII 2011, Part I. LNCS, vol. 6771, pp. 239–248. Springer, Heidelberg (2011)
16. Sleeman, D.: Umfe: a user modelling front-end subsystem. Int. J. Man-Mach. Stud., 71–88 (1985)
17. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)
18. Nazemi, K., Burkhardt, D., Breyer, M., Kuijper, A.: Modeling users for adaptive semantics visualizations. In: Stephanidis, C. (ed.) Universal Access in HCI, Part II, HCII 2011. LNCS, vol. 6766, pp. 88–97. Springer, Heidelberg (2011)
19. Luo, H., Niu, C., Shen, R., Ullrich, C.: A collaborative filtering framework based on both local user similarity and global user similarity. Mach. Learn., 231–245 (2008)
20. Gong, S.: A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software 5, 745–752 (2010)
21. Guo, L., Peng, Q.: A combinative similarity computing measure for collaborative filtering. In: Proceedings of ICCSEE 2013. Advances in Intelligent Systems Research, pp. 1921–1924. Atlantis Press (2013)
22. Brusilovsky, P., wook Ahn, J., Dumitriu, T., Yudelson, M.: Adaptive knowledge-based visualization for accessing educational examples. In: Tenth International Conference on Information Visualization, IV 2006, pp. 142–150 (2006)
23. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-weighted user model for recommender systems. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 97–106. Springer, Heidelberg (2007)