Visualization Adaptation Based on Environmental Influencing Factors

Dirk Burkhardt[1], Kawa Nazemi[1], Jose Daniel Encarnacao[1], Wilhelm Retz[1], and Jörn Kohlhammer[2]

[1] Fraunhofer Institute for Computer Graphics Research (IGD), Darmstadt, Germany
{dirk.burkhardt,kawa.nazemi,jose.daniel.encarnacao, wilhelm.retz}@igd.fraunhofer.de
[2] Technische Universität Darmstadt, Darmstadt, Germany
joern.kohlhammer@gris.informatik.tu-darmstadt.de

Abstract. Working effectively with computer-based devices is challenging, especially under mobile conditions, due to the various environmental influences. In this paper a visualization adaptation approach is described, to support the user under discriminatory environmental conditions. For this purpose, a context model for environmental influencing factors is being defined. Based on this context model, an approach to adapt visualizations in regards of certain environmental influences is being evolved, such as the light intensity, air quality, or heavy vibrations.

Keywords: Adaptive Visualization, Information Visualization, User-centered Interaction, User Experience, Sensor Fusion.

1 Introduction

The work efficiency and effectiveness with information visualization systems may depend on the user-centered adaptation capability considering various influencing factors. Therewith, human information retrieval systems with strong user-centered approaches provide sufficient support for the diversity of information-related tasks. The user-centered approaches are commonly developed as static systems supporting in particular specific stakeholders in various domains, e.g. analysts. The static approaches have various advantages for the specific and known users and tasks, but in domains with variety of users the "one-system-fits-all" approach does not support the user sufficiently.

Adaptive information and adaptive visualization approaches model users' behavior based on the implicit users' interactions or by identifying users' tasks focus in particular on such user-driven influencing factors. However, the effectiveness of information visualization may be influenced by further factors too. One of these indirect factors is the environmental influence. Today, the environmental influencing factors are commonly investigated in environments with extreme conditions, such as manufactories or computer systems that are used in dusty, noisy, very hot and very cold

M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2014, LNCS 8510, pp. 411–422, 2014.

environments. In perspective of the increasing relevance of mobile computing, the considerations of environmental issues get important for the everyday use too. In mobile environments a couple of factors have influences on users and their interaction behavior. Among others, common aspects are changes of light (bright conditions outside under the sun, dark in some rooms, which makes it hard to watch information on screens), vibrations (as they effect during travels in trains, which limit the precision of pointing interactions) or sticky air conditions (heavy weather in summer or bad air in smaller rooms, which limits the mental conditions of users and restricts the use, e.g. of complex visualizations, too).

There are approaches that adapt systems based on environmental aspects. Some basic approaches are also integrated in notebooks and smartphones, for instance the use of light sensors to change the brightness of the screen. However, these concepts are designed to adapt the general system on the environmental factors, but these concepts lose sight of adapting the visualizations based on the environmental factors. Also the transfers of environmental aspects to concrete adaptations of the visualizations are less regarded. But adapting visualizations in a more appropriate way to the environmental factors, and in a well configurable way, can increase the usability of visualizations and thus increase the efficiency of the user working with it.

Therefore, first of all this paper introduces different influencing factors that might be useful for adapting visualizations. To investigate the variety of possible influencing factors, we introduce a conceptual framework of environmental influencing factors. In this context, the most obvious sensor-types will be investigated and modeled to a "context model". This context model is the baseline of investigating further factors of influence, but refers in its first version on data about light intensity, temperature, humidity, air quality, and vibrations. In a proof of concept we consider these sensor data to adapt the visualization during a common search scenario.

2 Environment Based User-Interface Adaptation

To realize adaptive systems, it is required to consider the relevant information about the user and the context [1]. Meanwhile, many adaptation strategies do regard the user and his behavior as main actor on the technical system, whereas other aspects, such as the process or the environment as influencing factor, is less considered. But, the additional contextual information, such as GPS information from physiological sensors, can be valuable, as far as they can help to identify relevant information for the user in more concise manner [2,1]. To summarize, research on user-adaptive systems focused on variables corresponding to the categories: current state of user, longer-term properties of user, user's behavior with state, and consequences for user [1]. Whereas the main focus in the area of context-aware computing has been on the categories: readings from context sensors, readings from physiological sensors, and features of the situation [1].

The complexity of how many and which factors do influence a user can be very heterogeneous. This depends on a couple of aspects. Therefore Motti [3] defined a 7 dimensional context-aware design space that encompasses, among other, technical, environmental, process and user aspects. Malek et al. [4] defines the context from a generalized point of view and distinguishes between internal and external environmental factors on the highest level. He classified the user model and the current state during interaction as internal environmental factors. As external environmental factors, he classified physical (location, temperature, light etc.), temporal, hardware (bandwidth, processor speed, etc.), software (e.g. the used OS) and social (e.g. connected users) environmental aspects. Especially the adaptation capabilities of external environmental factors, more precisely the physical aspects, are in focus of our work. One representative is the system of Samulowitz et al. [5], which aims to adapt the user-interface and the presented information based on the current user location. Therefore they adapt primary on the data level to filter and highlight certain pieces of data. Most other systems and approaches focus on the adaptation of physical aspects in mobile environments, and more precisely on mobile devices, such as PDAs or smartphones [6,7]. In all of these adaptation approaches - with respect to environmental factors - they aim to adapt the entire user-interface, such as rotating the entire presentation, if the device is rotated, or changing the contrast and brightness, if the environment is very bright (bright sunshine) or dark (in close rooms with dimmed lamps). The used approaches, to adapt the entire user-interface, are currently also standard in notebooks and smartphones, but only on a very basic level with limitation on just one or two sensors.

To the best of our knowledge, there is no application or prototype available that aims to adapt concrete visualizations based on the environmental influence factors.

3 Visualization Adaptation Based on Environmental Influencing Factors

3.1 Overview

The adaptation needs to focus on two major parts. The first part is the most technical, because it concentrates on the sensor data acquisition and processing pipeline. The second part is a more conceptual one, it specifies how the visualization needs to be adapted based on the influencing factors.

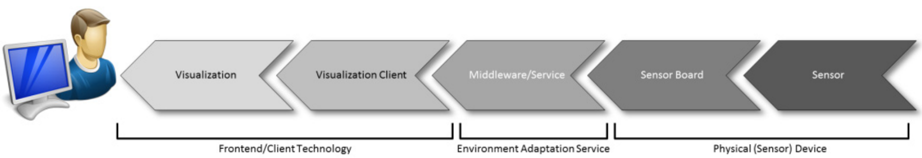

Fig. 1. The abstracted sensor data processing pipeline, starting from the sensors, where the data are captured to adapt the visualizations

To adapt visualization based on environmental influencing factors, it is essential to measure environmental constraints. Therefore it is necessary for the first part, to define a pipeline where sensors measure these constraints and transform and prepare them for the adaptation of the visualizations.

For this purpose, we defined a canonical processing pipeline for the processing of environment sensors, which is illustrated in Fig. 1. At the initial point, sensors (as part of a sensor board) are measuring a certain aspect, e.g. light intensity or noise level. Afterwards all the sensor data are collected and summarized at middleware service, which we mentioned as Environment Adaptation Service. Here, the data can be further prepared and will be provided through an API to frontend programs, which use these information to optimize the visualization based on the environmental factors.

3.2 The Environment Context-Model

Currently, a number of environment context-models exist [4,6,15]. For our concrete adaptation purpose, we focus on those models, which focus mainly on the environmental factors in a physical manner. As a baseline, we orient our specification primarily on the work of Malek et al. [4]. They distinguish between internal environmental factors, such user models and current states, and external environmental factors, such as physical, temporal, hardware, software and social environments.

As we aim to adapt on the application level, the applicable environmental influences are limited. Therefore, we reduced the major categories of Malek et al. [4] on physical and hardware factors. We extended their list of factors, because they provided only some examples. Our final environment context model is than specified as follows:

3.3 Architecture and Sensor Processing Pipeline

One major challenge in adapting visualization within programs is the collection of relevant sensor data. On mobile devices, for instance on modern smartphones, it can be easy, because they provide these information directly. On computers and notebooks it is more challenging, because only a very limited number of computers contain integrated sensors that measure the environment. Therefore, it is essential to provide a middleware to consider this issue and provide alternative interfaces for external sensor boards. These boards should not be limited on specific and established sensors, for example because the light intensity can be measured through a light sensor, the distribution in the histogram of a web-cam picture, indirect through the eye size, and user made contrast and brightness changes of the monitor. For each kind of input sensor, the middleware can define a possible environmental influencing factor.

In fact, the adaptation of the visualization has to follow a simple data procession pipeline (see Fig. 2), beginning with the sensors, for instance a temperature or light sensor, or in indirect form, e.g. through a camera or audio sensor as part of a webcam. These sensors are connected to a sensor board, for instance an Arduino that directly connects some sensors, or a webcam that includes a video and audio sensor. These sensor boards are connected to the middleware, we mentioned as Environment Adaptation Service. On this service, the sensor values will be stored and additionally used to generate abstract environment conditions, such as bright, humid or hot environment. This sensor data acquisition can be very easy, if just specific sensors are used, and can become more complex, if environment factors have to be calculated e.g. from a webcam picture. At the end, the server owns a couple of environment conditions and the raw sensor values.

Table 1. The environment influencing factor context (based on Malek et al. [4]). Each category consists of some environmental influencing factors that can be measured and used to adapt the visualization.

Category	Influence factors	Examples
Physical environment	Location and orientation	- Changes of light intensity, e.g. bright sun outside and dark rooms - Reduced air quality in rooms without air conditions
	Noise	
	Dirt	
	Weather	
	Temperature	
	Humidity	
	Pressure	
	Light	
	Air quality/concentration	
	Vibration	
Hardware environment	Device capabilities	- Changes of the resolution needs changes at the application - Support of special interaction devices, e.g. an 3D-Mouse to navigate to 3D Visualizations
	Bandwidth	
	Network capacity	
	Connectivity	
	Processor speed and supported processing features	
	Graphic card speed and supported processing features	
	Caching/Memory capacity	
	Storage capacity	
	Resolution	
	Sound Quality	
	Sound Power	
	Battery	
	Input Devices	
	Output Devices	

To adapt visualizations, there must be an application that allows considering such environment conditions in the visualization generation. Therefore, it needs a connection possibility to connect to the Environment Adaptation Service to request the environment conditions (or the raw data). To have a flexible API with extension ability for further adaptation strategies, we used a simple socket API, but it can also be integrated on a webserver as web-service. To consider the environmental influencing factors in a sufficient way during the visualization generation, sensor data processing should base on a stable processing concept.

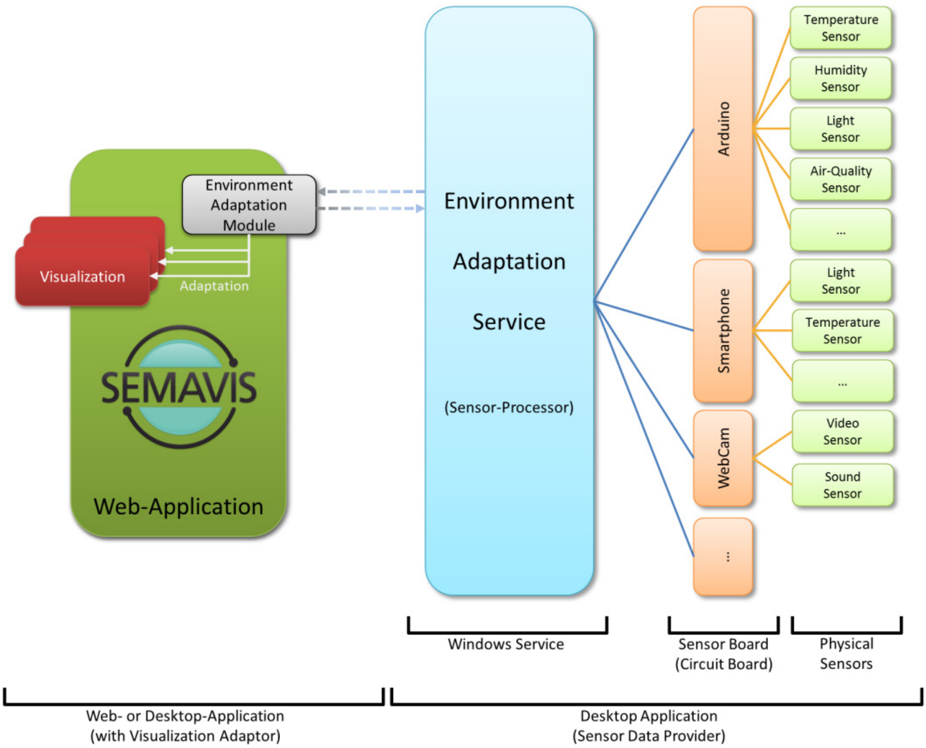

Fig. 2. Overview over the general architecture where the *Environment Adaptation Service* collects the values of different sensors from various sensor boards and processes environmental conditions. These conditions are provided to applications that use the data for adaptation purposes.

3.4 Adaptation Concept

In the previous section, the technical architecture was described, which is responsible to measure environmental and process it for the provision for the final application or program that needs to use this information for the adaptation purposes.

We have the data for the general visualization purpose at the visualization application, e.g. Linked-Open Data for visualizing the semantic information, and we have the environment condition data, which should be used to optimize the visualizations. Because of the fact that the sensor data are not that complex, we generally can confine the further processing strategies to adapt the visualization on two methods. The first methods are machine learning approaches. Machine learning approaches are useful, if the system should learn what conditions are appropriate for which changes/parameterization of the visualizations. The advantage would be that the threshold when and which graphical primitive etc. has to be changed, is set automatically on an optimal value. The second approach is a rule-based approach. Here, the parameterization will be applied based on some (static/fix) defined rules, e.g. if the light sensor value is higher than 300 lumen, then the contrast and brightness should be increased.

In our adaptation concept, we chose a rule-based approach, because training of the system gets very difficult, since we just have novice and advanced users, who use the system not regularly. Another advantage of the rule-based approach is the easier extension with further sensors, where only new rules for these new sensors have to be specified.

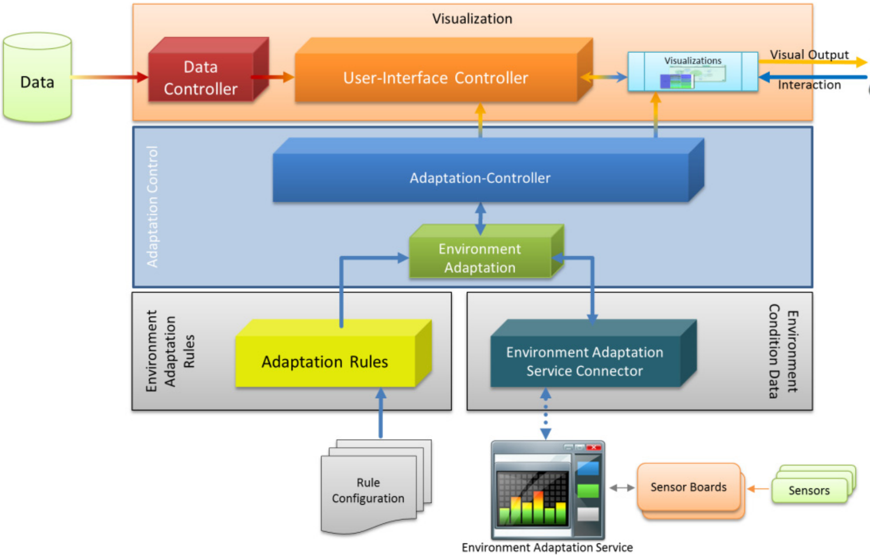

Fig. 3. This illustration shows how the adaptation will be applied on the visualization. Therefore, the adaptation rules from a configuration will be used and compared in the in the *Environment Adaptation* with the condition data from *Environment Adaptation Service*.

In Fig. 3 we show the internal adaptation processing. On the top we have the normal data processing for the visualization generation. On the bottom, we have a configuration with the concrete adaptation rules and the connection module to the Environment Adaptation Service, which provides the environment conditions. Based on both information, the *Environment Adaptation* generates the required changes on the user-interface and – more important – on the visualizations. The *Adaptation-Controller* has just a managing functionality, because our system supports a number of adaptation capabilities [11,12,13,14]. Such a certain *Adaptation-Controller* is not required if visualizations should only be adapted on environmental factors.

The generated changes on the user-interface and the visualizations were considered during the user-interface and visualization generation. Therefore it is essential that both parts provide the required adaptability.

3.5 Definition of Adaptation Rules

As mention above, the adaptations are specified in a static manner through adaptation rules. These rules can (1) address the entire user-interface, (2) visualizations of the same type, and (3) a concrete visualization. What adaptation should be performed in case of which environment factor change, needs to be defined by experts.

The application of adaptation on the entire user-interface can be useful, if the effect is not limited on a specific visualization. For instance, if the contrast and brightness of the general user-interface should be reduced, it is mandatory to change all visual elements, because one bright element on the screen can have the same effect as if all visualizations have a high brightness. We summarize adaptations on visualizations and the user-interface in general under this adaptation group, which can also include menus, buttons and many more. So, it is not limited on real information visualizations.

The adaptation of visualizations of the same type is reasonable, if visualizations with a similar visualization algorithm are used, e.g. node-graph visualizations. An example is the change of the level-of-detail, which only makes sense on visualizations allow showing different level of details. The adaptation of similar visualizations in only a single adaptation rule can also support to let those visualizations look similar in any situation. Users can have a better understanding of the presented data, instead of being confused, if the same data are visualized differently, because of different adaptation changes in fact of different rules.

To allow specific changes on certain visualizations, it is necessary that single visualizations can be adapted, too. This allows, among other things, changes on the general layout, e.g. instead of showing information in radial form they can be shown in a top-down way. The adaptation of single visualizations should be avoided, because if any visualization is adapted with its own bag of rules, the changes can look random and so they are an inefficient feature for the user. If, for instance, the one node-graph visualization changes category nodes in blue and another one in red, it only perplexes the user. Thus, it is recommended to use adaptation approaches, which taking care for the entire user-interface or for a group of visualizations of the same type. Rules for single visualization should only be used to enable or disable very visualization specific aspects.

3.6 Towards a Multi-adaptation Visualization System

Adaptation of visualizations is often a beneficial strategy. Most often the user is in focus and it is aimed to adapt visualization to the user factors. In the past, we have therefore conceptualized and implemented a couple of adaptation approaches [11,12,13,14], which take just a single aspect into account. Each of the presented approaches helps the user to work with the visualizations in a more efficient (solve tasks faster) or a more effective way (solve tasks more comfortable).

A crucial point is to enable more than one of these approaches at the same time, especially, if different adaptation methods affecting a visualization. To avoid unwanted effects in the visualizations, e.g. different node color changes because of user adaptation and environment adaptation, we currently allow only one adaptation technique at the same time. But this idea limits the possibilities, because the visualization can only be optimized by a single influencing factor.

Hence, we conceptualized an adaptation manager (the adaptation manager is equal to the adaptation controller in Fig. 3). The adaptation manager organizes all adaptation actions. So, all changes on graphical primitives, layout aspects etc. are notified to the adaptation manager. The adaptation manager stores all changes (by type and value) for any visualization (the name) and what adaptation technique has done that in a large list. These four properties build a quadruple and are stored as one entry in the list. In case that another adaptation technique wants to make similar changes in a single visualization, e.g. changing the color of a node, the entire change will be blocked, until the other technique revert the adaptation to the previous state. This approach avoids most unwanted curious states of visualization, because of many independent operating adaptation techniques.

4 Implementation

We applied the described concept through the integration of an Arduino (Fig. 4 shows the circuit), which connecting a number of sensors. We focused on physical environmental influencing factors for the prototype. By way of example, we included a temperature and humidity sensor pro to measure the environment temperature and relative humidity. We also integrated a light sensor to detect the ambient light intensity, and an air quality sensor to monitor over indoor air conditions. We tested a tilt switch, to detect vibrations. A small program runs on the Arduino that provides the sensor values over the USB connection to the computer.

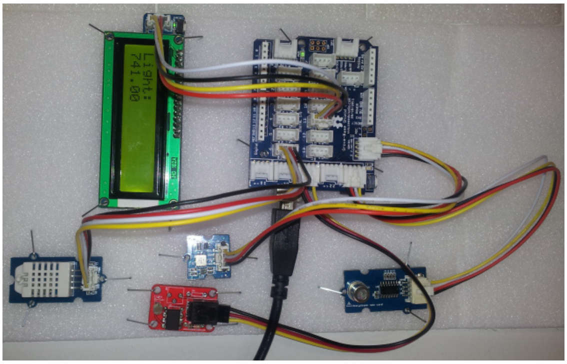

Fig. 4. To test our concept, we use an Arduino[1] with a couple of sensors that shares the sensor values with the Environment Adaptation Service. Instead of the Arduino Board also other kinds of sensor systems can be connected.

A simple Java program acts as our Environment Adaptation Service. It collects the sensor values and generates the environment conditions. Other applications can

[1] The Arduino is a programmable microcontroller that allows easily connecting various sensors and other kinds of electronic bricks for different use cases. More information on: http://arduino.cc (last accessed: 21/01/2014).

connect to this service and can request the environment conditions or the raw sensor values through a separate provided socket connection.

The adaptation functionalities are implemented in the web-application SemaVis, which is used in a common search scenario. SemaVis is a web-technology to visualize heterogeneous kinds of data. A separate module requests the current environment conditions and based of pre-defined rules, the visualizations will be adapted. In this very first implementation we only considered adaptations based on the light, which adapts the contrast and brightness of the visualization, on motions, which adapts the size of some nodes/buttons, and the air quality, which adapts the complexity of some visualizations. Fig. 5 shows the adaptation on the visualizations based on the light intensity and the air quality.

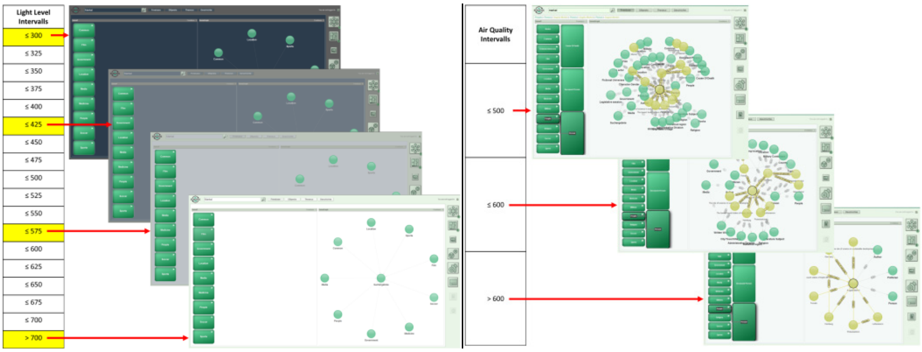

Fig. 5. On the left side, the screenshots show the changes of the contrast and brightness in dependence of the environment lightness. The darker the environment the darker the visualizations will be drawn. On the right side the visualization complexity (level of detail) is changed in dependence of the air quality. The worse the air quality of the environment is detected, the more reduced the complexity of the visualization.

5 Discussion

To validate our approach to adapt the visualization based on environmental influencing factors, we noticed the challenge of evaluating adaptive visualizations. In contrast to implementation of other features, e.g. to visualize some issues in an improved manner, it is more difficult to make a cross-evaluation. We expect the reason for this circumstance is that the provided visualization functionalities have not been changed. Even more, they are just refined slightly through an improved subjective perception. These subjective perceptions are mainly interpreted as User-Experience, and the evaluation of User-Experience is a complex topic [10,8]. In the most evaluation experiments the time is measured to compare (e.g. [9]), if users could solve certain tasks faster [10]. But, for the evaluation of the subjective perceptions such experiments will not be suitable. However, questionnaires can give feedback on personal perception, but whether this is also a practical and beneficial approach cannot be determined.

It is also critical how an evaluation should be realized in practice, because it is not so easy to change the environmental factors in a natural way. Simulated environment changes, e.g. vibrations or changes of the temperature in offices, give a quite surreal impression. To evaluate adaptations based on environment factors is a challenge, since it is mostly not perceived as something normal.

6 Conclusion

In this paper we introduced an adaptive visualization approach based on environmental influencing factors. For this purpose we investigated similar approaches in research and already existing implementations in the market, especially in mobile devices. Furthermore, we investigated the variety of possible influencing factors and defined a context model, which builds the baseline for further influencing factors. Based on this theoretic context model, we described a conceptual framework of an adaptation system that adapts visualizations based on the influencing factors.

In the first prototype we utilized data on light intensity, temperature, humidity, air quality, and vibrations. In a proof of concept, we considered these sensor data to adapt the visualization during a common search scenario.

Acknowledgments. Part of this work has been carried out within the FUPOL project, funded by the European Commission under the grant agreement no. 287119 of the 7th Framework Programme. This work is based on the SemaVis technology developed by Fraunhofer IGD (http://www.semavis.com). SemaVis provides a comprehensive and modular approach for visualizing heterogeneous data for various users.

References

1. Zimmermann, A., Specht, M., Lorenz, A.: Personalization and Context Management. User Modeling and User-Adapted Interaction 15(3-4), 275–302 (2005)
2. Jameson, A.: Modelling both the Context and the User. Personal Ubiquitous Comput. 5(1), 29–33 (2001)
3. Motti, V.G.: A computational framework for multi-dimensional context-aware adaptation. In: Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2011), pp. 315–318. ACM, New York (2011)
4. Malek, J., Laroussi, M., Derycke, A.: A Multi-Layer Ubiquitous Middleware for Bijective Adaptation between Context and Activity in a Mobile and Collaborative learning. In: International Conference on Systems and Networks Communications (ICSNC 2006), p. 39 (2006)
5. Samulowitz, M., Michahelles, F., Linnhoff-Popien, C.: Adaptive interaction for enabling pervasive services. In: Banerjee, S. (ed.) Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDe 2001), pp. 20–26. ACM, New York (2001)
6. Schmidt, A., Beigl, M., Gellersen, H.-W.: There is more to Context than Location. Journal of Computers and Graphics 23, 893–901 (1998)

7. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing techniques for mobile interaction. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000), pp. 91–100. ACM, New York (2000)
8. Arhippainen, L., Tähti, M.: Empirical evaluation of user experience in two adaptive mobile application prototypes. In: Proceedings of the 2nd International Conference on Mobile ldots (2003)
9. Van Velsen, L., Van der Geest, T., Klaassen, R., Steehouder, M.: User-centered evaluation of adaptive and adaptable systems: A literature review. Knowl. Eng. Rev. 23(3), 261–281 (2008)
10. Vermeeren, A.P.O.S., Lai-Chong Law, E., Roto, V., Obrist, M., Hoonhout, J., Väänänen-Vainio-Mattila, K.: User experience evaluation methods: current state and development needs. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI 2010), pp. 521–530. ACM, New York (2010)
11. 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.) HCII 2011, Part I. LNCS, vol. 6771, pp. 239–248. Springer, Heidelberg (2011)
12. Nazemi, K., Burkhardt, D., Praetorius, A., Breyer, M., Kuijper, A.: Adapting User Interfaces by Analyzing Data Characteristics for Determining Adequate Visualizations. In: Kurosu, M. (ed.) HCD 2011. LNCS, vol. 6776, pp. 566–575. Springer, Heidelberg (2011)
13. Burkhardt, D., Ruppert, T., Nazemi, K.: Towards Process-Oriented Information Visualization for Supporting Users. In: IEEE Education Society: ICL 2012: 15th International Conference on Interactive Collaborative Learning and 41st International Conference on Engineering Pedagogy. IEEE, Inc., New York (2012)
14. Burkhardt, D., Nazemi, K.: Dynamic Process Support Based on Users' Behavior. In: IEEE Education Society: ICL 2012: 15th International Conference on Interactive Collaborative Learning and 41st International Conference on Engineering Pedagogy. IEEE, Inc., New York (2012)
15. Castillejo, E., Almeida, A., López-de-Ipiña, D.: User, Context and Device Modeling for Adaptive User Interface Systems. In: Urzaiz, G., Ochoa, S.F., Bravo, J., Chen, L.L., Oliveira, J. (eds.) UCAmI 2013. LNCS, vol. 8276, pp. 94–101. Springer, Heidelberg (2013)