2012 International Conference on Cyberworlds

Interactive Exploration System: A User-Centered Interaction Approach in Semantics Visualizations

Dirk Burkhardt, Christian Stab, Martin Steiger, Matthias Breyer, Kawa Nazemi
Fraunhofer-Institute for Computer Graphics Research
Darmstadt, Germany
{dirk.burkhardt, christian.stab, martin.steiger, matthias.breyer, kawa.nazemi}@igd.fraunhofer.de

***Abstract*—Nowadays a wide range of input devices are available to users of technical systems. Especially modern alternative interaction devices, which are known from game consoles etc., provide a more natural way of interaction. In parallel to that the research on visualization of large amount of data advances very quickly. This research was also influenced by the semantic web and the idea of storing data in a structured and linked form. The semantically annotated data gains more and more importance in information acquisition processes. Especially the Linked Open Data (LOD) format already experienced a huge growth. However, the user-interfaces of web-applications mostly do not reflect the added value of semantics data. This paper describes the conceptual design and implementation of an Interactive Exploration System that offers a user-centered graphical environment of web-based knowledge repositories, to support and optimize explorative learning, and the integration of a taxonomy-based approach to enable the use of more natural interaction metaphors, as they are possible with modern devices like WiiMote or Microsoft Kinect. Therefore we introduce a different classification for interaction devices, and current approaches for supporting the added values in semantics visualizations. Furthermore, we describe the concept of our IES, including a strategy to organize and structure today's existing input devices, and a semantics exploration system driven by user-experience. We conclude the paper with a description of the implementation of the IES and an application scenario.**

Keywords: User-Centered Systems, Semantics Visualization, Multimodal Interaction, Semantic Web

I. Introduction

Semantic-Web and semantically annotated knowledge and information gain more and more importance in future information and knowledge acquiring processes. While formal descriptions of information, e.g. Ontology or Topic Map are still under the investigation of research institutions and enterprises, semantic structures based on the collective intelligence of web-users became silently an inherent part of the web. Especially the Linked Open Data (LOD) format has experienced a huge growth in the open internet and became an established data model for conceptualizing knowledge entities and describing semantic relationships between knowledge entities and domains. The Linked Data format is not only used to model a specific domain by a small set of knowledge engineers, it is rather a reflection of the knowledge interpretation of a whole community, which models domain-knowledge for structuring and disseminating to a diversified audience. A single Linked Data database gains millions of knowledge entities per day and grows faster than experts have anticipated.

Existing semantics visualization techniques do not consider the surpluses of the Linked Open Data structures, where the semantics structure has to be built-up with a routine of query requests. They focus on various but specific ontology characteristics, e.g. displaying the hierarchical inheritance structure, multiple inheritance or semantic relations between ontology entities. The complex structure of the Linked Data varies, based on the users' query on the data. The heterogeneity of the requested data should be exploited for the visualization and hence enable a more efficient interaction with the underlying semantics.

Offering the use of modern interaction devices on technical systems is an actual trend considering the customers' needs to provide an easy and intuitive interaction. Especially games are predominantly designed to work with other controllers than traditional ones like joystick, gamepad or the combination of mouse and keyboard. In 2006, the Nintendo Wii was published and had an amazing success because of its intuitively useable control paradigm that allowed for performing natural gestures. Going one step further, Microsoft's Kinect introduced a full body gesture interaction by using the player's body as a controller device. But not only systems for playing games try to make use of natural interactions; also systems like mobile phones using multi touch are successful like Apple's iPhone or other modern smartphones. All of these systems have one feature in common: they support a natural interaction through supporting gestures.

On the computer nowadays a gesture-based interaction is still not very successful. On these systems the traditional interaction devices mouse and keyboard are the preferred devices to control applications. Only multi-touch monitors are in some usage scenarios used for an easier interaction e.g. in public domains. Above all, the reason for that is the missing support in programs and applications. But also if alternative interaction devices are supported, their usage may not be adequate in all use case scenarios of a program. In different use cases, different interaction metaphors are needed to provide a useful interaction. For example, when presenting a picture, abstract gestures are useful to instruct the image-viewer application to zoom or rotate the picture.

978-0-7695-4814-2/12 $26.00 © 2012 IEEE
DOI 10.1109/CW.2012.45

But if the user navigates through the menu of such a program only simple gestures are appropriate like pointing at an entry or panning the display in a direction.

In the last years, both processes, i.e. the research on graphical visualizations as well as the research in alternative interaction techniques, experienced a rapid development. Nevertheless, up to now, no adequate connection between them exists. One of the goals of this paper is to develop a method for using alternative and more intuitive interaction-devices e.g. the WiiMote in web-based graphical visualization. Within the range of this development, there are further aspects, which have to be taken into consideration. Intuitive use is a subjective perception of a person [1], so the user's perceptions must be integrated into the development-process (see Figure 1). But considering the users individual perception is time-consuming, which most developers cannot or does not want to offer.

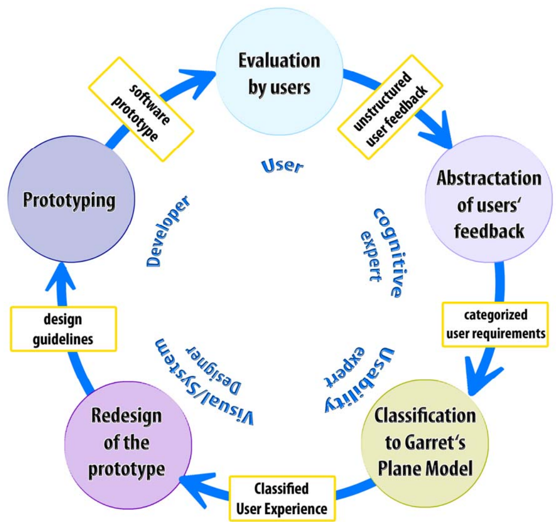

Figure 1. Process Model of User-Centered Software Design [1]

II. Related Work in User-Centered Interaction

In both areas, the research on graphical visualizations as well as the research on alternative interaction techniques were underlying significant changes in the past. But there are some major aspects that are important to consider, if the graphical visualization should be able to operate with alternative interaction techniques.

A. Classifications for interaction devices

In the past, some effort was put forth in defining classifications of input devices, mainly with the goal of a better understanding of the devices, so that optimal devices could be found for specific tasks or similarities in devices to be able to replace a device by different, but compatible one [2].

One of the first classifications was a taxonomy, which was defined by Foley et. al. [3]. It was focused on graphical user interface applications and their typical tasks that can be performed. Tasks were, for example selecting a graphical object, orienting the mouse etc. Foleys taxonomy differentiates between devices that can perform these tasks and in the way they do it, especially if the active principle is direct or indirect.

Buxton et. al. [4] made the observation that there is a major difference in the way devices can produce the same output. Buxton called these differences pragmatics and divided the devices further by the way they produce the output (position, motion, pressure, etc.) and ended up with a taxonomy, which could represent simple devices.

Finally, in order to be able to classify all input devices including virtual ones, Mackinglay [5] took Buxton's taxonomy and build a formal approach, which can be used to classify the most input devices. Mackinlay describes an input device with 6-tuples:

These tuples can be combined by 3 different merge operations forming more complex input devices:

- Connect: connects one device from "Out" with a fitting input device from "In"
- Layout composition: indicates the position for two simple devices at a superordinate device
- Merged composition: similar to layout composition, but the merged values define a new complex data type

It is difficult to insert devices for video or audio-based recognition in these taxonomies. In an alternative approach Krauss [6] divides devices on a higher level in coordinate-based and non-coordinate-based interaction devices. Thus, his classification is driven by the output values of the devices.

B. Gesture-based interaction systems

Some effort was invested into support for a gesture-based interaction to normal computers and into application for allowing an easier control. In particular, the use of interaction devices from game consoles gained interest for the use on regular computers. The reason is that they are well known and apart from their easy usage, they provide different methods for interacting with a system. Early interaction devices like old data gloves were designed for only one interaction scenario. In contrast, , a modern controller like the WiiMote, an input device developed by Nintendo, can recognize acceleration in three dimensions. Thus, it can be used for gesture-recognition, but also to use the infrared camera as pointing device or use the buttons directly to control an application. The WiiMote can be used on a computer using a one of manydifferent available implementations. One of the most flexible programs for using the WiiMote as gesture-based device is WiiGee. The recognition is implemented by a statistical comparison of the acceleration inputs and comparing them with previously trained acceleration data. WiiGee can classify the observed sequence as gestures [7], [8].

There are also video-based applications that recognize faces [9] and some early approaches to recognize human gestures in video-based systems [10]. A more advanced approach is Microsoft's Kinect which uses stereo images to recognize gestures.

Speech recognition is a field that gets more and more involved and works well for a limited vocabulary. Many systems like cellphones, car navigation systems, operation systems and computer games support speech recognition.

C. *Semantics Visualization*

Semantic visualization is not a new field. The investigation of the massive semantically structured data on the Web poses new demands for visualizing the semantic structure and requires facing the problems of information overload. In this section we introduce two semantic visualization systems, which both use the approach of combining different visualization techniques to improve overview of structure, relations and detail of the data and give a short introduction in our previous work of the combination of visualizations in a knowledge Cockpit [11, 12].

Knoocks (Knowledge Blocks) [13, 14, 15] is a desktop-application for the visualization of ontologies, which combines several visualization techniques for semantic data. To display concepts of a specific ontology, Knoocks uses the so called ‘Knowledge Block’ approach. A block is used to visualize a concept and includes all of its subconcepts within the block-boundaries. Besides the block-visualization of the structure, you can also use the outlook-window and a node-linked visualization component to view the semantic relation of the data.

Thinkbase [16, 17] is a visualization and exploration tool for the Freebase database, which uses the Freebase UI as part of its interface. It consists of two combined views of the same data, a graph-visualization of the relational structure and the text-based Freebase interface, which shows detailed properties of the topic of interest. Next to it, there is also a graph-visualization window that shows all instances, which are directly related to the topic of interest and combines all relations with the same role into so called ‘aggregation nodes’ [16] to provide a better overview. The second part of the interface is the Freebase-UI, which contains textual information, images and other properties and relations of the selected topic.

Also, this combined visualization has some advantages in giving the user the possibility to understand the neighboring structure of the instance in focus and simultaneously provide information about its properties. The system also deals with problems in adaptability and lucidity. The user cannot decide whether or not he wants to see more than the direct neighbors of the instance in focus and every time he navigates through the graph, the structure changes completely, so he has to adapt to the new look in every step. Moreover, the Freebase-UI is not designed to be a part of a combined visualization, so it shows too much data, needs a lot of scrolling and is not interacting well with the graph visualization.

As described in our previous work, the Visualization Cockpit [11, 12] is a user-adaptable approach of visualization combination. It lets the user combine different visualization techniques and representation details (like color, icons, depth of relational structure) into a personal ‘Knowledge Cockpit’. Thus he can display the same data in different visualization techniques, which focus on different aspects of the data to get a deeper understanding of the whole semantics structure. The user may, for example, combine the SeMap-Visualization [18] with a graph-like visualization to combine the advantages of the quick and easy overview capability of SeMap with the more detailed, but complex visualization of the graph.

III. Concept of the Interactive Exploration System

In the first phase we have to differ between the two required parts of a user-centered interaction system for semantics visualization. It consists of an interaction system, which manages the input devices and transforms all user interactions into technical commands that control the visualization. This includes the technical sampling of the data-stream that is supplied by the devices and the transformation into an action within the user-interface. In contrast to classical interaction devices like the mouse, most of modern interaction devices are based on gestures, so that also approaches which are able to support the recognition of performed gestures have to be considered. Therefore there is a need for integration possibilities of gesture recognition techniques which will be implemented in modules. Furthermore, such gesture-based systems include two parts with the main part being the recognition component. The second part is some kind of learning-system, which enables the system to learn new gestures (Figure 2). This is complemented by the visualization part, which addresses the adequate visualization of the data. To support intuitive use and individual search strategies, such systems have to provide modern search strategies in visual user-interfaces, which are primarily supported in visualization for semantically annotated data. This processing pipeline will be described in this chapter.

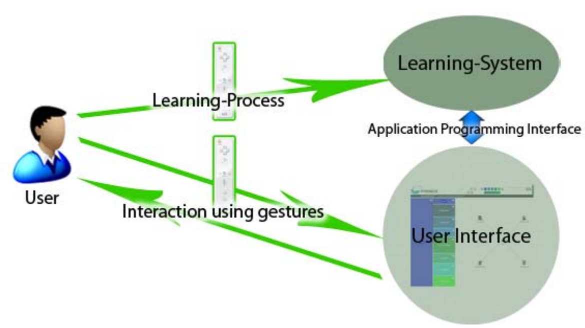

Figure 2. General Interaction with the System

A. *Device Classification*

If a set of different interaction devices should be supported, it is useful to define a classification for the devices first. This will allow for an easier selection and support of the devices. In the past, a couple of such classifications were developed to group the existing devices mostly by their main characteristics; an overview was presented in the previous chapter. Regarding the implementation into technical systems the use of these classifications is limited. To overcome this limitation, we have developed a new classification taxonomy to classify input devices [19]. The classification is driven by two factors, (i) the interaction method e.g. a touch-based device and (ii) the interaction result e.g. the coordinates on the screen. Every device has to be classified in both aspects. It can also be represented as matrix, and the existing devices can be mapped onto the respective cells.

The first classification tree is about the *Interaction Methods* and can be subdivided into:

- Controller-based Interaction
  - Move-based
  - Touch-based
  - Acceleration-based
  - Button/key-based
  - Sensor-based
  - Camera-based
- Controller-less Interaction
  - Sensor-based
  - Camera-based
  - Microphone-based

The second classification tree regards the outcome. So the *Interaction Results* can be subdivided into:

- Non coordinate-based
  - Direct-Reaction / RAW-Data
  - Voice-Recognition
  - Posture-Recognition
  - Gesture-Recognition
- Coordinate-based
  - Indirect operation principle
    - 1-dimensional
    - 2-dimensional
    - 3-dimensional
    - n-dimensional
  - Direct operation principle
    - 1-dimensional
    - 2-dimensional
    - 3-dimensional
    - n-dimensional

The presented classification now allows for classifying every device by its used interaction metaphor and its result of interaction. But every device can also be grouped multiple times, for instance the WiiMote is able to be used as pointing device (sensor-based, controller-based interaction; 2-dimensional, indirect operating, coordinate-based results) and as gesture-based device (sensor-based, controller-based interaction; non coordinate-based, gesture recognition results). The concrete use of a device depends on the technical possibilities, because not every mode can be supported by an input device or perhaps the integrated sensors are not precisely enough to support some modes as required.

B. *User-oriented Interaction devices and Interaction Methods*

To organize the interaction device, we conceptualized an interaction system. The main work load is performed on the server. This includes the management of the devices, the interpretation of the interaction and the handling of the accessing application. It furthermore integrates the classification as a general model to group the devices. Therefore every device is embedded as a virtual device. So all required changes to support the handling of a concrete device as virtual device have to be considered within the capability-making process. But in consequence every existing device can be supported by the interaction system in the same way with the same accessing methods. The virtual devices can now be grouped to an interaction method and an interaction result. The processing of the device data will be done on the server. Because of the capsulation in a virtual device, most analysis modules can be used for all devices in the same group. This enables also the integration of new devices in a group without the need to invest effort to create a new analysis module.

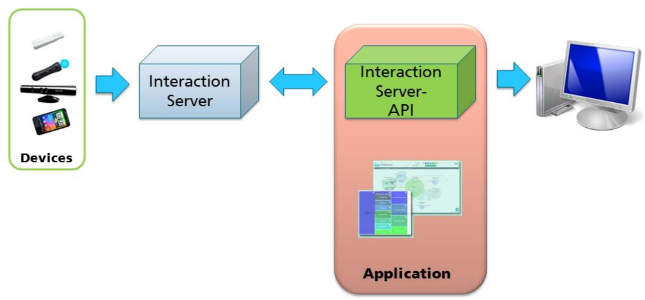

Figure 3. Architecture model of the Interaction System

Applications are accessing the input device indirectly through the Interaction Server-API. Through the API the applications are able to select a specific device, device group (depending on the interaction method and the required interaction result) or mix of device groups. The entire data process will be performed on the server and only the results of the analysis will be sent back to the application. This reduces also the required effort for the integration of the interaction system.

To support user-oriented interaction, the interaction system forms the basis. But for a full user-oriented interaction the application has to make use of the features of the system. Thus, it has to consider that kind of devices that will make most sense to work within the application and for the task that needs to be performed. It is also recommended to support more than one group of interaction devices to interact, so that the user can switch, e.g. if the WiiMote does not work well for the user, he needs the possibility to switch to the mouse or touchscreen as alternative interaction forms.

C. *Semantics Visualization*

To visualize semantically annotated data we are using the SemaVis-framework [20]. One advantage of the framework is the defined data processing pipeline that supports most features of semantic data. For a user-centered visualization there was a so called "Knowledge-Cockpit" developed, which allows the user to orchestrate his preferred view on the data [12]. The view can consist of a set of different types of visualization e.g. a node-link graph visualization and a tree visualization. Another advantage is the available freedom in using the system. There are many ways for interacting with the graphical user-interface, which makes it highly interactive.

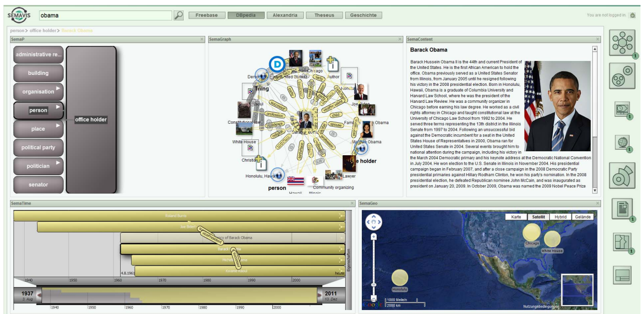

Figure 4. Screenshot of the SemaVis-Cockpit (adapted from [12], [20])

D. User-Centered Interaction in Semantics Visualizations

The named parts of interaction system and semantics visualizations are building the concept for the Interactive Exploration System. It allows the interaction through semantic data-sets with alternative interaction devices like the WiiMote, Playstation-Move controller or Kinect. Especially in large-scale displays these alternative forms of interaction are allowing a more intuitive interaction and can enable a better user-experience than the traditional interaction with computers with mouse and keyboard. Moreover, the general support of additional input devices allows for the expansion with devices that are designed for the interaction in graphical user-interfaces. In the first step, only pointing and gesture-devices are supported for the interaction in the semantics visualization. The pointing functionalities are used to interact in the data. This encompasses the selection of instances and concepts from the semantic data. The gestures are limited to simple direction gestures to right, left, up and down. These gestures are used for selecting visualizations and move them to a different position.

This limitation is necessary, because the navigation in graph visualization etc. can be difficult and complex. To support an intuitive navigation through data and visualizations, the supported metaphors must be understandable. Maybe there are other kinds of interaction metaphors which could be more intuitive or are better understandable, but there is currently less research invested in this topic and thus further studies are required to investigate it.

The advantage of the combination is also in supporting the virtual learning and knowledge exploration through a positive user experience-driven interaction. So it can be more attractive especially for younger people to develop knowledge.

IV. The Interactive Exploration System

In order to integrate the interaction system in SemaVis, we have implemented the API. Through the integrated API it is possible to define the allowed input devices and its interaction form and interaction result. Regarding the interaction itself, we are using a large-scale screen to make the pointing on a specific data entitiy easier. Our primary interaction device was the WiiMote, because it allows for the integration as pointing device as well as gesture-based interaction device. In normal operation the WiiMote is used as a pointing device. To select an entity, the B-button has to be pressed. If the A-button is pressed and is held, a selection is created based on the entities under the cursor. The user can move to the next or previous visualization by performing a direction gesture (left, right, etc.) . If the A-Button is double-clicked, the visualization can be moved to a different position on the screen. By pressing the A-Button again, the last position of the visualization will be set as the final position. In this manner the user is able to create his Knowledge Cockpit.

Figure 5. The exploration with a modern interaction-device in semantics visualization

Additional implementations of input devices are planned. Currently, we are working on an implementation of the Playstation Move controller and Microsoft Kinect. In this paper the general approach of a classification-driven interaction system that is integrated and coupled with semantics visualization was in the focus to show the benefits of such a resulting interactive exploration system.

V. Use-Case: Multimodal Interaction in Semantics Visualization

The first implementation, which is described in this paper, is applied in semantics visualization. The main goal was to provide an alternative interaction form to access the data, which is more attractive for users. Because of the better experience that users might have, the exploration through knowledge becomes more interesting. Especially younger people can be attracted more, if they have the possibility to work with knowledge over such natural interaction interfaces.

In a next step we are working on a novel approaches to use visualization in the policy-modeling processes. To allow for a more comfortable and perhaps interesting analysis, we plan to try this system also in such a political environment. We expect that this approach can help to make e-participation interesting and attractive also for younger people, who are, often enough, not interested in political topics.

VI. Discussion

The paper excludes some issues that needs be discussed or studied further. The topics are primaryly addressing the multimodal interaction in graphical visualizations. An important challenge here is the gesture-based interaction in graphs. This addresses the gesture interaction in graphs in general e.g. how and which gestures can be used to control the navigation in graphical visualizations, and in detail. One of these detail question is the mapping of direction-gesture to visualized neighboring nodes. The navigation by gestures is critical, because the implementation can be done in many different ways. So, among other, the direction can be measured in degrees and mapped to the best fitting node in the visualization. But it is also possible to orient on the available links of the selected node.

Another challenge is the selection of the current focus area. In most application there are different areas e.g. for the menu, visualization and parameterization. An open question is how to switch between these areas, if non coordinate-based interaction forms are used. One possible solution would be to structure the layout like it is in treemap-visualizations. Then, the direction to an area can be clearly identified, but in complex user-interfaces this approach will fail.

VII. Conclusion

In this paper we presented a new approach to use modern interaction devices that provide a more natural interaction way in semantics visualization. We therefore introduced a newly created classification, which allows for classifying the type of interaction metaphor and with respect to the result of the interaction. The classification was then evaluated in an interaction system, which organized all available input devices and the analysis of the generated interaction data. The results, e.g. a recognized gesture, are supplied to the application over an easy-to-use API. For our described IES we used this API in combination with a semantics visualization system, the SemaVis-framework. SemaVis allows for navigation through semantic data-sources by different combinations of graphical visualizations. This allows for the creation of personalized views on the data and is called “Knowledge Cockpit”. The integration of alternative interaction devices permits additional freedom in navigating through the data and makes the interaction more attractive and intuitive, because the most interaction parts are performed by gestures, which are orienting on human’s gestural interaction.

The preliminary results of a use-case study were throughout positive. The interaction was for most people more attractive. The reported positive user-experience was also positive, which indicates that it was motivating especially younger people to explore through knowledge databases to gather new information. These results can be considered when new learning strategies will be developed for younger people and pupils.

References

[1] Nazemi, K., Hornung, Ch.: Intuitive Authoring on Web: a User-Centered Software Design Approach. In: Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008, pp. 1440-1448. AACE, Vienna (2008)

[2] R. J. K. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen, Jr., Integrality and separability of input devices. ACM, 1994.

[3] J. D. Foley, V. L. Wallace, and P. Chan, The human factors of computer graphics interaction techniques. Prentice Hall Press, 1990.

[4] W. Buxton, Lexical and pragmatic considerations of input structures. ACM, 1983.

[5] J. Mackinlay, S. K. Card, and G. G. Robertson, A semantic analysis of the design space of input devices. L. Erlbaum Associates Inc., 1990.

[6] L. Krau, Entwicklung und Evaluation einer Methodik zur Untersuchung von Interaktionsgerten für Maschinen- und Prozessbediensysteme mit graphischen Benutzungsoberflächen. Universität Kaiserslautern, 2003.

[7] B. Poppinga, Beschleunigungsbasierte 3D-Gestenerkennungmit dem Wii-Controller. University of Oldenburg, 2007.

[8] N. H. S. B. Thomas Schlomer, Benjamin Poppinga, Gesture Recognition with a Wii Controller. 2nd International Conference on Tangible and Embedded Interaction, 2008.

[9] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld: Face recognition: A literature survey. ACM Comput. Surv., vol. 35, pp. 399-458, December 2003.

[10] S. Mitra and T. Acharya, Gesture Recognition: A Survey. IEEE SYSTEMS MAN and CYBERNETICS SOCIETY, 2007.

[11] Nazemi, K., Breyer, M., Burkhardt, D., Fellner, D. W.: Visualization Cockpit: Orchestration of Multiple Visualizations for Knowledge-Exploration. In: International Journal of Advanced Corporate Learning (iJAC), Vol 3, No 4 (2010)

[12] Nazemi, K., Breyer, M., Burkhardt, D., Fellner, D. W.: Semantic Visualization Cockpit: Adaptable Composition of Semanitcs-Visualization Techniques for Knowledge-Exploration. In: Conference ICL2010, (2010)

[13] Kriglstein, S., Motschnig-Pitrik, R.: Knoocks: New Visualization Approach for Ontologies. In: Information Visualisation, 2008. IV'08. 12th International Conference, pp.163--168, IEEE (2008)

[14] Kriglstein, S., Wallner, G.: Knoocks - A Visualization Approach for OWL Lite Ontologies. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp.950--955, IEEE (2010)

[15] Kriglstein, S.: OWL Ontology Visualization: Graphical Representations of Properties on the Instance Level. In: Information Visualisation (IV), 2010 14th International Conference, pp.92--97, (2010)

[16] Hirsch, C., Grundy, J., Hosking, J.: Thinkbase: A Visual Semantic Wiki. In: Demo Session of the 7th International Semantic Web conference, (2008)

[17] Hirsch, C., Hosking, J., Grundy, J.: Thinkbase: Interactive Visualization Tools for exploring the Semantic Graph of large Knowledge Spaces. In: Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009), (2009)

[18] Nazemi, K., Breyer, M. and Hornung, C.: SeMap: A Concept for the Visualization of Se-mantics as Maps. In: Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction (UAHCI '09). Part III, Springer, pp. 83-91, (2009)

[19] Burkhardt, D., Breyer, M., Glaser, C., Nazemi, K., Kuijper, A.: Classifying Interaction Methods to Support Intuitive Interaction Devices for Creating User-Centered-Systems. In: Proceedings of Universal Access in Human-Computer Interaction. Design for All and eInclusion. Berlin, Heidelberg: Springer, LNCS 6765, pp. 20-29., (2011)

[20] Nazemi, Kawa ; Stab, Christian ; Kuijper, Arjan: A Reference Model for Adaptive Visualization Systems. In: Jacko, Julie A. (Ed.): Human-Computer Interaction: Part I : Design and Development Approaches. Berlin, Heidelberg, New York : Springer, 2011, pp. 480-489.