# Modified Virtual Reality for Intuitive Semantic Information Visualization

Katja Einsfeld, Achim Ebert
Computer Science Department, Technical University Kaiserslautern, Germany
einsfeld@informatik.uni-kl.de, ebert@informatik.uni-kl.de

Jürgen Wölle
Institute of Urban Water Management, Technical University Kaiserslautern, Germany
woelle@rhrk.uni-kl.de

**Abstract**

*In order to make use of domain knowledge and thus to create less abstract visualizations of the available information, we propose to integrate information visualization techniques in 3D visualizations of the application subject. We describe a framework that is build on an ontology, an hierarchical visualization toolkit, and a modular concept of modeling 3D layouts. This allows us to integrate semantically related information directly in the scene. We achieve this integration by modifying the real world geometry in several ways. Our concept helps non-expert users to intuitively interact with the system and to understand what is going on.*

*Keywords*—**Process Data Visualization, Semantic Data Visualization, Virtual Reality**

## 1 Introduction

The KOMPLETT-project ("komplett" is German for "complete") is engaged with the development of a small unmanned Waste Water Treatment Plant (WWTP) with two distinct water circles and resource recycling ambitions. The final goal of the information visualization system for this complex plant is the intuitive representation of large amounts of heterogeneous data (e.g. process data, optimization suggestions, expert knowledge, maintenance instructions) depending on the user profile. The visualization should provide detailed and revealing information for the remote expert as well as intuitive access to relevant instructions for the non-expert on site. In this context multiple semantic perspectives are essential. The development of the WWTP as well as the visualization system are still in progress.

## 2 Related Work

In the scientific community there is little research on SCADA(Supervisory Control and Data Acquisition)-systems. There are uncountable publications on abstract graph visualizations for processes, information hierarchies, and knowledge visualizations. For example Bosca et al. [2] propose an abstract ontology visualization called OntoSphere3D. Due to its 3Dness, this application allows some degree of immersion, that is possible with only abstract spheres and lines in the scene. Research concerned with real world geometry, e.g. Virtual Reality applications are in most cases restricted to the visualization and animation of the virtual scene for specific application areas like virtual manufactoring (Jin et al. [8]).

A common practice in information visualization is the utilization of real-world metaphors for abstract data. In the example of Web Forager and Web Browser by Card et al. [3] there was even a semantic similarity between metaphor and data.

There are, however, only rare examples in which information visualization techniques are combined with real-world geometry visualization of the application area: Kirner et al. [9] placed some information visualization elements like circle diagrams, bars, text elements, and metaphors like books in a walk-through environment. The goal was to visualize the behavior of visitors of a museum in the real environment of that museum. Jern et al. [7] developed an information visualization component toolkit that consists of graphs, bar charts, parallel coordinates, scatter plots, and also less abstract elements like geographical maps and 3D layouts. The individual components are linked to each other and use the same color scale. Thus, abstract data can be related to physical layouts although the 2D window-concept prevents direct connection. In previous work [5] we already used real world 3D metaphors like a shelf of books to represent documents. Additionally to this approach, which might remind of Card's Web Forager, we used information visualization techniques to represent document meta-data: The books color, size, and their position in space were directly used to represent various properties and document similarities.

Current industrial process visualizations approaches like WinCC, InTouch, or the open source pvbrowser start to benefit from 3D visualizations. The simultaneous sensible use of information visualization techniques and the semantic relation of elements of the 3D visualization to additional information like documentation are, however, still rare. Moreover, there are still a lot of usability concerns due to the fact, that these systems often have a long historical background, that can not easily be abandoned. Another problem of many current SCADA-systems is the gap between the abstract data and the real physical layout of the plant.

## 3 Motivation

Concerning our experiences with abstract process data visualization and evaluating feedback from the users of these first approaches, we decided to move our concepts and information visualization techniques to real-geometry representations. Users did not only wish for more realistic and intuitive representations, they also stated that it would not be enjoyable having to switch between various visualization modes during work. Even semantic data experts, that usually concentrate on abstract representations, told us that, according to their experience, the visualization of semantic data (e.g. ontology-visualization) should make use of the knowledge available in the respective domain and therefore be less general.

Thus, the goal of our research was to create a flexible 3D visualization based on the real geometry of our WWTP and providing and integrating semantic information in this perspective.

## 4 System overview

The presented 3D information visualization is called HANNAH, which is short for "Here And Now, Near At Hand". The name accounts for the claim for creating a visualization- and HCI-framework which is context-sensitive, adaptable, ergonomic, intuitive, vivid, allows direct manipulation, and provides direct animated feedback.

The concept of our system is as follows: The heterogeneous data from multiple sources and for multiple purposes are collected on one ontology. The ontology is modeled with the Open Source ontology editor Protégé[1] in collaboration with domain experts and information scientists. The information in the ontology is organized in more than 160 classes, 1600 instances, and nearly 20.000 relations or attributes from 125 diverse slots. To give an example, there is a class that contains all physical parts of the plant organized in subclasses. The instances of this class have attributes like label and description. They have relations to instances from the same class (e.g. preceding and succeeding part of the plant) and from other classes (e.g. manufacturer, documentation). This way of relating information items to each other adds semantics and context information that allow a huge range of flexible utilization. There is external data like pdf-documents, images, videos, 3D objects, web-pages, and numerical process data in databases referenced from the ontology.

To allow the program uniform and fast access to all this information, the ontology is exported in rdf-format and than converted into database representation with the help of the Open Source tool rssdb[2]. The advantage of this approach is, that all kind of applications from php-scripts to OpenGL-programs can access the semantic information using simple SQL-queries.

Our framework is implemented in C++/OpenGL and contains, besides class hierarchies and compounds for visualization objects, classes that can be used as local data structures to represent parts of the ontology that are relevant in the current context: CommonGraph holds lists of CommonNodes and CommonArcs that model the instances and relations from the ontology. CommonNodes and CommonArcs store their types, attributes, and relations to each other. The CommonGraph structure can then be used to generate visual graph representations.

CommonGraph also provides methods that encapsulate SQL-queries and thus allows us to easily collect all instances of a specific type or to add some needed attributes to a list of instances. Besides other possibilities, CommonGraph can generate user defined mappings between two lists of nodes. For example, the user might want to view all temperature data collected in the process and relate it to the process step in which it was measured. To collect the required data, CommonGraph is first asked to generate a list of all process steps $A$. Then an other list containing references to process data items with the additional condition that the items should be related to temperature measuring instruments (i.e. temperature items) is created $B$. Finally, a mapping between $A$ and $B$ is generated by giving CommonGraph the additional information that the mapping should be of the following semantic quality: $A$ has to *involves-plant-part*-relate to plant part $C$ and $C$ has to *has-information*-relate to $B$. CommonGraph return mappings from $A$ to $B$ and from $B$ to $A$ in the form of C++ STL map-containers that can efficiently be used in the visualization.

In the subsequent sections, there will be multiple examples for ontology enhanced visualizations that will further clarify the benefits of our ontology based concept.

[1] http://protege.stanford.edu/

[2] http://139.91.183.30:9090/RDF/RSSDB/

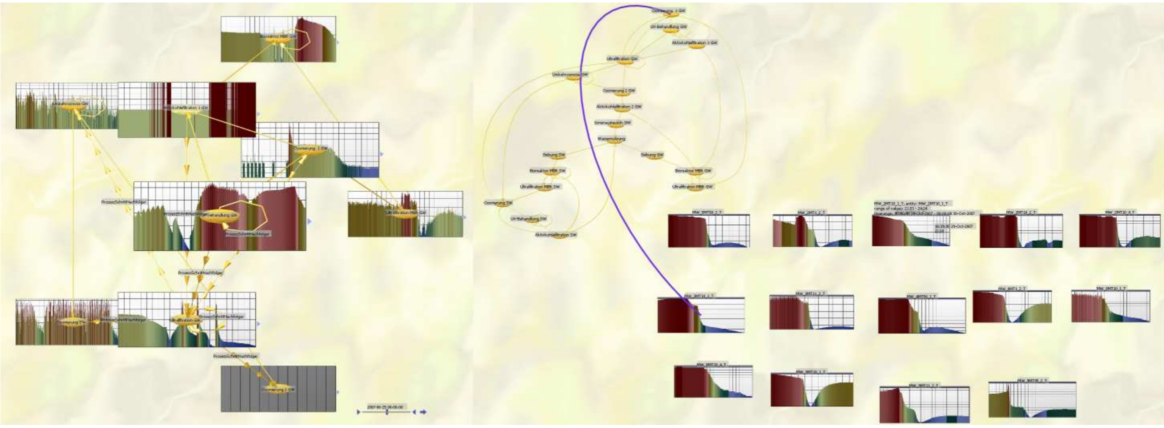

Figure 1: Two versions of abstract process data visualization

# 5 Process Visualization

## 5.1 Abstract Process Visualization

With the goal of gaining flexibility compared to the static process views common in most current process visualization systems, we started visualizing the process logic with automatically layouted graphs generated by graphviz.

Two examples of these rather abstract graphs are shown in figure1. Depending on the current context, extracts of the semantic structure are read from the database representation of the ontology into a graph data-structure and visualized with the help of the Open Source graphviz library[3]. The nodes can also be moved to CircleMode (figure 1, left), where the focus node is in the front- and center-position and the connected nodes are organized on two circles around this node. The transition between the graphviz-overview perspective and the CircleMode focus perspective is animated. The arrows between the nodes indicate the flow direction in the process (or an other semantic relation depending on the current configuration). They are emphasized when the cursor is over connected nodes by moving multiple arrows along the curve. This was reported to be very helpful by users browsing the graph-structure.

Additionally we displayed semantic information in the form of semitransparent labels directly in front of the nodes. Process data semantically connected to the process node was visualized in multiple ways: First (figure 1, left), we attached the diagrams directly to the nodes. Nodes that have multiple diagrams attached allow clicking through these diagrams one after the other. Then (figure 1, right) we experimented with a visualization that uses separate visualization areas for diagrams and process graph and visually connects a node and diagram if they are semantically connected and one of them is clicked. While in the first approach the diagram-process-steps-connection is more intuitively perceived by avoiding the two distinct areas, the second approach avoids occlusion.

In first informal evaluations test users stated that these abstract visualizations can be useful for understanding the logic of the process. For most daily tasks as checking the plant status or locating the source of misbehaviour they would prefer more realistic visualizations. This is especially important for non-experts. Thus, we decided to create additional real-geometry perspectives for process visualization.

## 5.2 Real Geometry Process Visualization

Some drawbacks of current real geometry plant visualizations are that they (1) do not add additional information, (2) important information is hidden due to occlusion, or (3) the user is "lost in space" when having to navigate through the 3D world. Our approach tries to solve the three problems by (1) semantically relating parts of the real world geometry to additional information, (2) using flexible hierarchical visualizations and combining 2D and 3D visualizations, and (3) disburdening the user from having to move in the 3D world.

The third point is achieved with the help of the *Thought Wizard* metaphor derived from the theater metaphor which was described by Dachselt [4] and previously described by us [5]. The idea is to use simple interaction techniques to reorganize the geometry in the scene instead of having to concentrate on navigating through a fixed scene.

In order to achieve (1) and (2) we chose a more complex approach than simply modeling the geometry as a whole: First, we modeled every single component with the Blender-software. Then we stored the information relevant to construct the plant form the single components as it can

[3]http://www.graphviz.org/

be seen in figure 2 and 3 in a database. Finally, we imported this data in a subclass of our visualization framework.

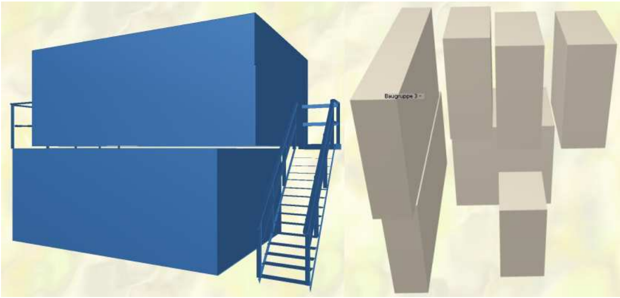

Figure 2: First and second level of hierarchical plant visualization.

Our data structure needed for modeling the geometry in the database consists of four tables: The first table stores for each physical element expect the tubes their position, orientation, and scale factor relative to the hierarchical mother-element, the material ID, and filename of the 3D object. The IDs from this table are referenced from the ontology, which allows the system to provide semantically related information for the physical components. The second table organizes these elements in an hierarchical order by storing IDs of mother- and daughter-elements. E.g. the outside view of the plant seen in figure 2, left side, has one entry in this table of each of its daughter-elements, the logical plant parts shown in figure 2, right side. One logical plant part contains numerous functional elements and some of these are further subdivided in several components. The third table lists all tubes and tube elements with their position, material ID, filename of 3D object (if and only if they are not composite tubes), and connectivity data. Connectivity data consists of five columns that contain the default value or the ID of connected objects. The first two of these columns contain IDs of connected plant components (the ones stored in the first table). The last three columns contain IDs of other tubes to which this tube might be connected. The last table stores information about the hierarchical construction of tubes from the basic components (straight tube, corner tube, T-tube, ...). Each entry in this table contains hierarchical information, length (for straight tubes), position, and orientation relative to the hierarchical mother. This design allows heavy reuse of simple 3D components which allows efficient visualization and, at the mean time, provides a level of abstraction due to the hierarchical concept. Thus, the semantic connectivity information needs only to be stored for logical tube components and not for each straight and corner element they consist of.

In our visualization framework, every 3D object that is loaded is stored in a look-up-table so that other visualization objects can point to and use the same geometry. The user can explore the 3D geometry by rotating, translating, and scaling (mouse interaction) or by changing the hierarchy-level (plus- and minus-button). The options used to display and integrate additional information in the 3D view, that will be described in the next section, can be invoked by pressing the space bar to show the context menu and than selecting one entry.

Figure 3: Third level of hierarchical 3D plant visualization

Using real geometry is, however, not in every case the best practice: For example the second level of the hierarchical plant visualization (bounding boxes of logical parts) shown in figure 2, right side, has not proven to be useful. This might be due the occlusions or to the fact that the concept of logical parts is neither real-world nor intuitive to non-experts.

### 5.3 Information near at hand

In contrast to augmented reality (AR), the concept of HANNAH is more immersive as it brings heterogeneous data from various sources together in one homogeneous interface. In augmented reality systems information items like text are simply copied over the view of the real world. Thus the two media stay separated as they are from diverse quality and the user processes them in diverse ways. In contrast, the concept of HANNAH uses realistic 3D models of the real world in order to be able to manipulate this view to fit the current needs of the user and thus to integrate additional information in intuitive and heterogeneous ways. Moreover, HANNAH is more flexible than augmented reality systems: it neither depends on the movements of the user nor on techniques like head-mounted-displays and motion tracking. Instead its intention is to move the user's attention and not requiring the user to move her- or himself.

The HANNAH concept also contrasts to augmented virtuality (AV) as it does not simply augment the virtual representation of reality with additional information but rather integrates information in the virtual environment. Moreover, it even modifies the virtual representation in order to convey the semantics of what is perceived and thus to allow the user to gain knowledge of the underlying concepts and processes. (A taxonomy of mixed reality approaches was provided by Milgram et al. [10].)

Our approach of ontology based visualization allows us to display additional information about the component under the mouse pointer: Depending on expert or non-expert mode the technical code or the description of the element can be visualized as label. Additionally it is possible to directly link pdf-documentation, images, videos, and process data in the form of diagrams to a component. Thus the user can simply click on related relevant information near at hand.

Figure 4 shows the result of a user query for the connection between two components. This kind of visualization is extremely useful when learning about the operating mode or having to debug the plant, when only the problem is assumed to be somewhere between two components. This feature is implemented with the help of a variation of Dijkstra's path-finding algorithm. We considered plant parts (first table) as well as tubes (third table) to be "nodes" and used the connection data contained in the third table to move from one node to the next.

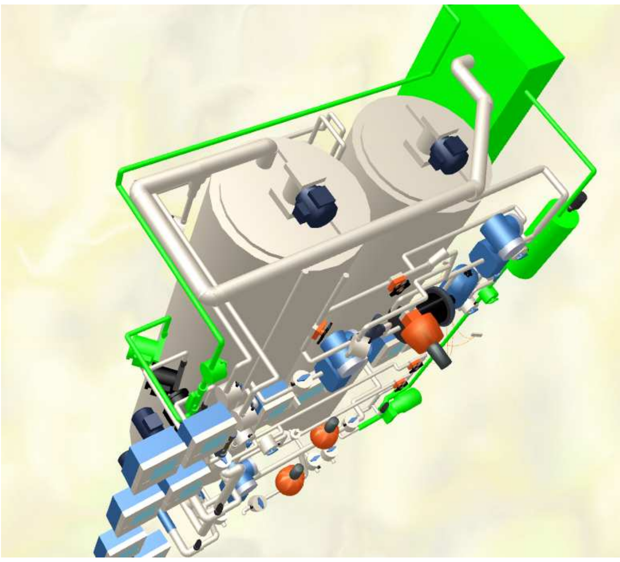

Figure 4: Detecting a path between two elements

Highlighting parts of the plant can also used in other cases: When the system detects an error, the defect component can be marked. The user can search for a component with the expert-signature "AB3XY12" or (with the help of our ontology representation in DB and SQL-LIKE) for components containing "pump" in their name and the system will mark the 3D objects representing the results.

In order to facilitate the visual search for marked objects, which might be critical in a problem situation, we chose green a signal color. According to Bauer et al. [1] this is a good choice in our case as green is not in the convex hull of the other plant-visualization-colors (CIE color space).

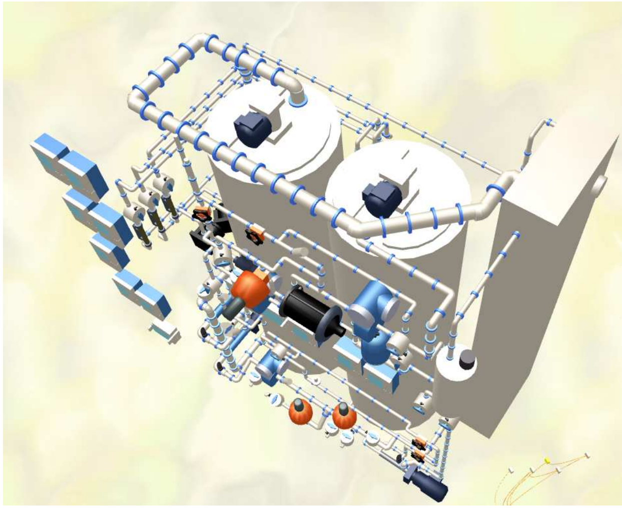

Figure 5: Indicating flow direction with animation

Figure 5 shows an animated visualization that indicates the flow direction of water or air in the pipes which can be toggled on and off. Here, animation and colored as well as modified pipe-geometry is used to help the user to understand the processes that are going on.

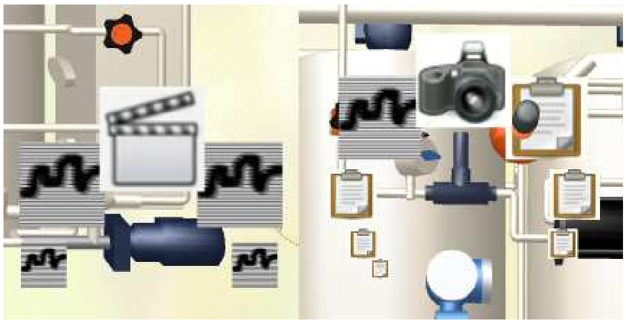

Figure 6: Two extracts from geometry with icons for additional information

Figure 6 indicates how icons can be attached to the 3D geometry to give the user the possibility to access additional information like process data in the form of diagrams, pdf-documents, photos and videos by clicking the icons. The positioning of the icons in space is not trivial, as it should be clear to which object they belong, they should however, not collide with the object, and they should not occlude too much. We try to achieve this goal by arranging the icons on focus-cirles around the objects and viewing only the icons of selected 3D objects.

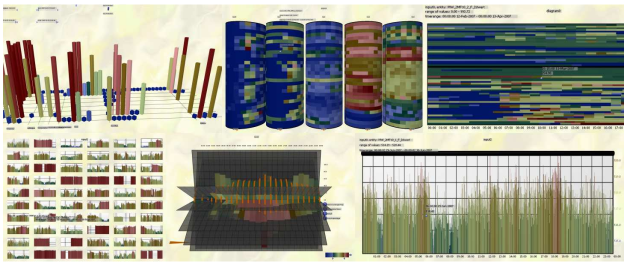

Figure 7: Diagram types: from overview to detail

Figure7 shows a set of diagram types available in the HANNAH-framework. A detailed description of the diagram concepts and some of the used metaphors can be found in previous publications [6]. One of the concepts is to make use of animation to solve the time-multiplexing- vs. space-multiplexing dilemma. We currently work on additional possibilities in directly integrate these metaphors in the real-world geometry.

## 6 Conclusions

Realistic 3D visualizations of physical plant layout help non-expert users to intuitively interact with the system and understand what is going on. While mixed reality systems (e.g. AR and AV) juxtapose real objects and virtual objects or abstract information, the HANNAH- concept integrates the two worlds in one homogeneous and intuitive interface.

In this paper we described how an ontology as an information basis combined with a modular construction of real world geometry can be used to integrate information in this scene. We propose to modify virtual reality in order to enrich the scene with additional information depending on the current task. Real-word modifications include hierarchical extracts, modified color, highlighting, animation, modified geometry, and attached icons.

Future work will involve research for new concepts of integrating diagrams and other information items into real-world geometry (e.g. color coding or texturing) and visualizing maintenance instructions with the help of flexible VR visualizations. Moreover, we will further evaluate the pros and cons of modified real-world visualizations and the pros and cons of integrating information vs. organizing it in a separate area with visual links to the first area.

In order to minimize the gap between physical layout and abstract data we additionally work on similar applications for mobile devices. With the help of these it will be possible to walk around the physical plant and simultaneously be navigated through the modeled plant and the relevant information.

## 7 Acknowledgements

This research is part of the interdisciplinary project KOMPLETT, which is supported by the German Federal Ministry of Education and Research (BMBF).

## References

[1] B. Bauer, P. Jolicoeur, and W.B. Cowan. Convex hull test of the linear separability hypothesis in visual search. *Vision Research*, 39:2681–2695(15), 1999.

[2] Alessio Bosca, Dario Bonino, and Paolo Pellegrino. Ontosphere: more than a 3d ontology visualization tool. In *SWAP*, 2005.

[3] Stuart K. Card, George G. Robertson, and William York. The webbook and the web forager: an information workspace for the world-wide web. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 111–ff., New York, NY, USA, 1996. ACM Press.

[4] Raimund Dachselt. Action spaces - a metaphorical concept to support navigation and interaction in 3d interfaces. In *Workshop 'Usability Centred Design and Evaluation of Virtual 3D Environments'*, 2000.

[5] Katja Einsfeld, S. Agne, M. Deller, A. Ebert, B. Klein, and C. Reuschling. Dynamic visualization and navigation of semantic virtual environments. In *IV '06: Proceedings of the conference on Information Visualization*, pages 569–574, Washington, DC, USA, 2006. IEEE Computer Society.

[6] Katja Einsfeld, Achim Ebert, and Jürgen Wölle. Hannah: A vivid and flexible 3d information visualization framework. In *IV07: 11th International Conference Information Visualisation*, 2007.

[7] Mikael Jern and Johan Franzen. Integrating infovis and geovis components. In *IV '07: Proceedings of the 11th International Conference Information Visualization*, pages 511–520, Washington, DC, USA, 2007. IEEE Computer Society.

[8] Li Jin, Zhigang Wen, and Ilias A. Oraifige. Distributed vr for collaborative design and manufacturing. In *IV '07: Proceedings of the 11th International Conference Information Visualization*, pages 792–797, Washington, DC, USA, 2007. IEEE Computer Society.

[9] Tereza G. Kirner and Valéria F. Martins. Development of an information visualization tool using virtual reality. In *SAC '00: Proceedings of the 2000 ACM symposium on Applied computing*, pages 604–606, New York, NY, USA, 2000. ACM.

[10] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *SPIE Telemanipulator and Telepresence Technologies 2351*, pages 282–292, 1994.