ADVANCED USER INTERFACES FOR PRODUCT MANAGEMENT SYSTEMS

Fredrik Gundelsweiler and Harald Reiterer
Human-Computer Interaction Workgroup, University of Konstanz
Universitätsstrasse 10, Box D 73, D-78457 Konstanz, Germany
{fredrik.gundelsweiler, harald.reiterer}@uni-konstanz.de

ABSTRACT
Few of today's EPDM (electronic product data management) systems make use of valuable approaches in user interface design and information visualization as suggested by researchers. In this paper, we describe a design approach addressing the problems of searching, browsing, visualizing and filtering information in hierarchically structured graphs. The main problem areas we identified are the amount of data, the possibly complex hierarchical structure in combination with a chronological versioning system and last the collaboration between different users. Working with such data spaces it is hard for users to keep a good overview on the one hand and to narrow the information space as required by the user tasks on the other hand. Additionally, the design of interaction and filter concepts for complex hierarchical networks remains a great challenge. We propose a design concept that makes use of different visualization and interaction techniques based mainly on overview and detail, filtering and zoomable components. In solving the visualization, search and filter problems of such PLM/EPDM data spaces we take a step forward to a fully optimized product lifecycle and the integration.

KEYWORDS
EPDM, parts documentation, user interface design, interaction techniques, zoom, overview and detail.

1. Introduction

A very little amount of today's information systems make use of valuable approaches in user interface (UI) design and information visualization as suggested by researchers. By example of an electronic product data management (EPDM) system, we present a case study in creating an advanced interactive system with help of such methods. First, we will introduce the EPDM project and explain its most relevant properties in Section 2 and 3. We will then deduce requirements for an advanced UI design solution. We describe the complex hierarchical network data space, the main user tasks and our design principles. Finally, we present our EPDM prototype in Section 4. It is based on a combination of overview & detail (O+D) techniques and zoomable components. Our hypotheses is that, with our system, data exploration and management is designed in a much more usable way compared to conventional EPDM systems. Using the example of a test data set, we show how user tasks can be supported by our ideas. In future we would like to conduct a usability study to identify further potential improvements concerning interaction and interface design to support the user's understanding of the information space. In Section 5, we summarize our findings and provide an outlook on our further research.

2. EPDM Systems and the Product Lifecycle

EPDM systems are applications for handling text, images, computer-aided design, interface diagrams etc. to document and relate technical products and their subcomponents.
Today modern companies use EPDM throughout their whole product lifecycle (PLM) to optimize their processes [14] (from the concept phase to the point of service and support). The general goals of EPDM are (1) the reduction of documentation and change management effort through more transparency, (2) the reduction of customer's costs for repairs, (3) the decrease of repair effort and (4) the reduction of risk with regards to international product liability. Despite the importance of EPDM, there are only a few PLM systems available, for example *Enovia*[1], *MatrixOne*[2], *mySap PLM*[3] and *TeamCenter*[4]. One problem is that these PLM tools are overloaded with features to support all target users throughout the whole product lifecycle. Therefore PLM systems are very complex and difficult to adapt to specific requirements. In this paper we focus on the problems of data visualization and interacting with such applications. Common problems in these systems are the lack of a

---

[1] IBM *Enovia*, demo system at http://www-306.ibm.com/software/applications/plm/enovia/demo/

[2] 3DS *Enovia MatrixOne*, http://www.matrixone.com

[3] *mySAP PLM*, http://www.sap.com/germany/solutions/business-suite/plm/index.epx

[4] UGS *TeamCenter*, http://www.ugs.com

concept for displaying the chronological relationships of object versions, an insufficient visualization of the product "is-part-of" relationships and inadequate navigation and interaction concepts. There are some ideas for new approaches to EPDM applications like [19] or [20] but they deal mostly with the system architecture than with the interaction and visualization of the interface. We did not find any related work that attempted to solve problems of EPDM systems by using filter, interaction and visualization techniques as described in current research.

Navigation is mostly implemented in the same way as in Microsoft’s Windows Explorer, using an expandable tree to navigate the hierarchical structure. If there are child objects with more than one parent, the tree navigation has to include these child nodes at multiple positions. The web interfaces of these systems are particularly lacking in interactivity and visualization, and even in giving users a clue as to where they are, where they can go to, and what they have already explored, not to mention more complex dependencies.

The analysis of conventional EPDM systems reveals essential weaknesses. Often there is no overview and no adequate narrowing of the information space. Therefore users have to accept a longer loading time because all data is loaded into the tree navigation. User tasks such as creating new component versions are supported by clicking through complex dialog steps in a wizard-like manner while the user loses task context. It is usually not possible to access task-relevant information while stepping through dialogs. Other poor solutions are static result displays, bad filter options and pop-up windows. The problem of displaying chronological and hierarchical data is often not addressed at all. Some of these problems can be solved by including the tree, graph or network visualization and interaction techniques described in current research papers (see Section 4.6). Our approach, however, makes use of a hierarchically structured graph layout combined with semantic zooming and dynamic queries. The goal of the EPDM project we describe in this paper is focused on improved usability for an easier access to the information space. We therefore break up with conservative UIs we have seen in current tools and demonstrate the value of advanced UIs. As a constraint, the application has to run as intranet application in order to be ubiquitously available and to avoid rich clients that rise more installation rollout and maintenance effort.

3. Basics, Constraints and Goals

In this section we want to explain some basics and constraints concerning the data space used. Although we think that there is a wide range of use for our system concept, the data space is rather special and further explanation is needed to understand it. The users of the system are working mainly with textual data describing the properties of component versions such as names, versions, voltages, and hardware manufacturer, with the possibility of some attached schematics and so on. The sample data space we are working on contains no multimedia data, but with our concept it should be possible to include it. It should be noted that we want to focus on an overall concept of the display, navigation and filtering of complex information structures, and not on data input masks.

All UIs of software applications and tools for the work with product data throughout the product lifecycle have the above mentioned problems. In solving the visualization, search and filter problems of such PLM/EPDM data spaces we take a step forward to a fully optimized product lifecycle and the integration and management of all product-relevant information. Possibly our interface concepts are applicable in other domains with similar data spaces, too.

3.1 Complex Hierarchical Data Space

Apart from the complexity there are some preconditions for data spaces to be used with our concept. It must be possible to build clusters within the network data and the hierarchical main structure must not be too complicated. Data must be summable - this is usually the case when hierarchical clustering can be used on the network data. To make filter operations possible, another precondition is that the objects of a cluster or hierarchy level must have the same attributes or dimensions. "Nice to have" are ordinal attributes, which make the filtering of data easier.

3.2 Hierarchy

The main structure consists of a hierarchy built upon “has-a” relationships. The complex hierarchical network data space we are working on is best explained by a simplified figure showing the overall relationships (“has-a”) between objects.

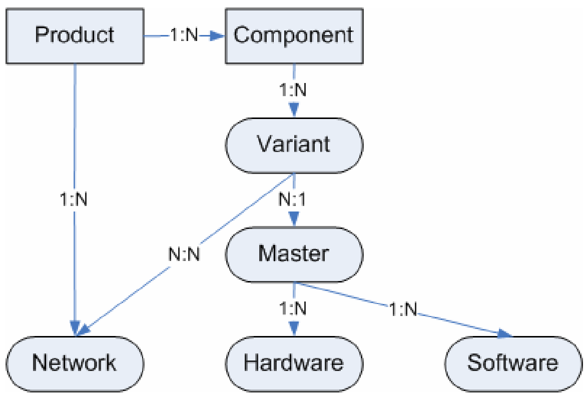

Figure 1: Simplified data model.

In Figure 1, a product consists of 1 to N components, and a component has 1 to N variants. At this point the cardinalities become inversed, because a master object is used to bundle variant data. Therefore more than one variant can be related to the same master object but a master object needs to be related to at least one variant. The master then has hardware and software components. Hardware and software entities consist of further components which are not shown in Figure 1 to keep it simple. The network is related to the variant object. Several variants can share one network and two networks can share one variant.

This composition seems relatively easy, but a lot of information is missing from the figure. We simplified the model by removing some objects and relationships to be able to concentrate on the main aspects. For example, the child objects of hardware versions are not listed and "is-compatible"-relationships between hardware and software versions are not included.

3.3 Chronological Data

The versioning of objects expands the data-space complexity by the dimension of time. New object versions can be created on the basis of existing object versions. As work on objects and their versions proceeds, more and more chronological dependencies emerge.

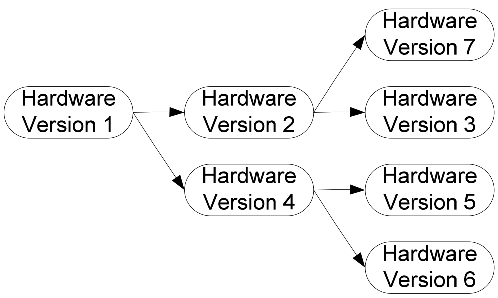

Figure 2: Branching of logical and chronological versions

Figure 2 shows a small example of the branching of hardware versions. The user created version 2 based on version 1, for example, and version 3 based on version 2. Version 4 will be technically based on version 1 and subsequently versions 5 and 6 are based on 4. To give another example, version 7 is based on version 2. The chronologically latest version is #7 and the logically latest versions are #3, 5, 6 and 7. Altogether, this leads to hierarchical and highly networked data structures.

3.4 User Tasks and Complexity

In the requirement phase we conducted interviews and focus groups with users, technical staff, and stakeholders. As a result our requirements analysis brought to light 6 major user tasks essential to EPDM (see Table 1). The very basic, but – given the complexity of the information space – none the less important user task is navigation to a product-part of interest (1). As some users might favour teleporting instead of browsing like described in [16], the UI has to offer search (2).

Table 1: Typical use cases relevant in EPDM

#	Task Description
1	Navigate to component versions
2	Search for component versions
3	Create a new component version
4	Check and adapt dependencies of versions
5	Compare different versions
6	Edit an existing version object

If the version is editable, users can make their own changes, otherwise the version is final and users have to create a new version to amend the relevant component data (3). After creating a new version, the user has to perform the task of checking and amending the relationships and dependencies (4). The creation of a new component version may lead to a chain of new component creations.

For example, the creation of a new software version can make it necessary to create corresponding hardware, master and variant versions as well (see Figure 1) and therefore additionally the check of existing relationships. Other tasks are searching for component versions (4) with specified properties or visualizing dependencies (5) such as the generally latest, or logically latest, versions (see Section 2.1). A comparison of component versions (6) and their relationships can help users to solve and complete their tasks (for example, the creation of a new version) in different scenarios. The chronological object versions may trigger a user's interest in their hierarchical relationships.

3.5 UI Requirements

Taking into account common UI tools, the information space, and the user tasks that we analyzed, we identify three critical areas for the UI design (see Table 2).

Table 2: General demands on EPDM advanced UI [18]

#	Topic	Demand on UI
1	Information Visualization	Visualize information space and relationships with overview plus detail techniques; let users define the degree of visual complexity
2	Navigation	Provide advanced navigation concepts; maintain context of use
3	Search	Provide search, filter and result presentation techniques

Concerning information visualization (1), there is a need to provide both O+D and F+C. The overview must provide the big picture of product-part relationships. The detail view must visualize the interdependencies of version objects. Whenever the user is interested in further information, a drill-down reveals more of the required product data, while other information is minimized.

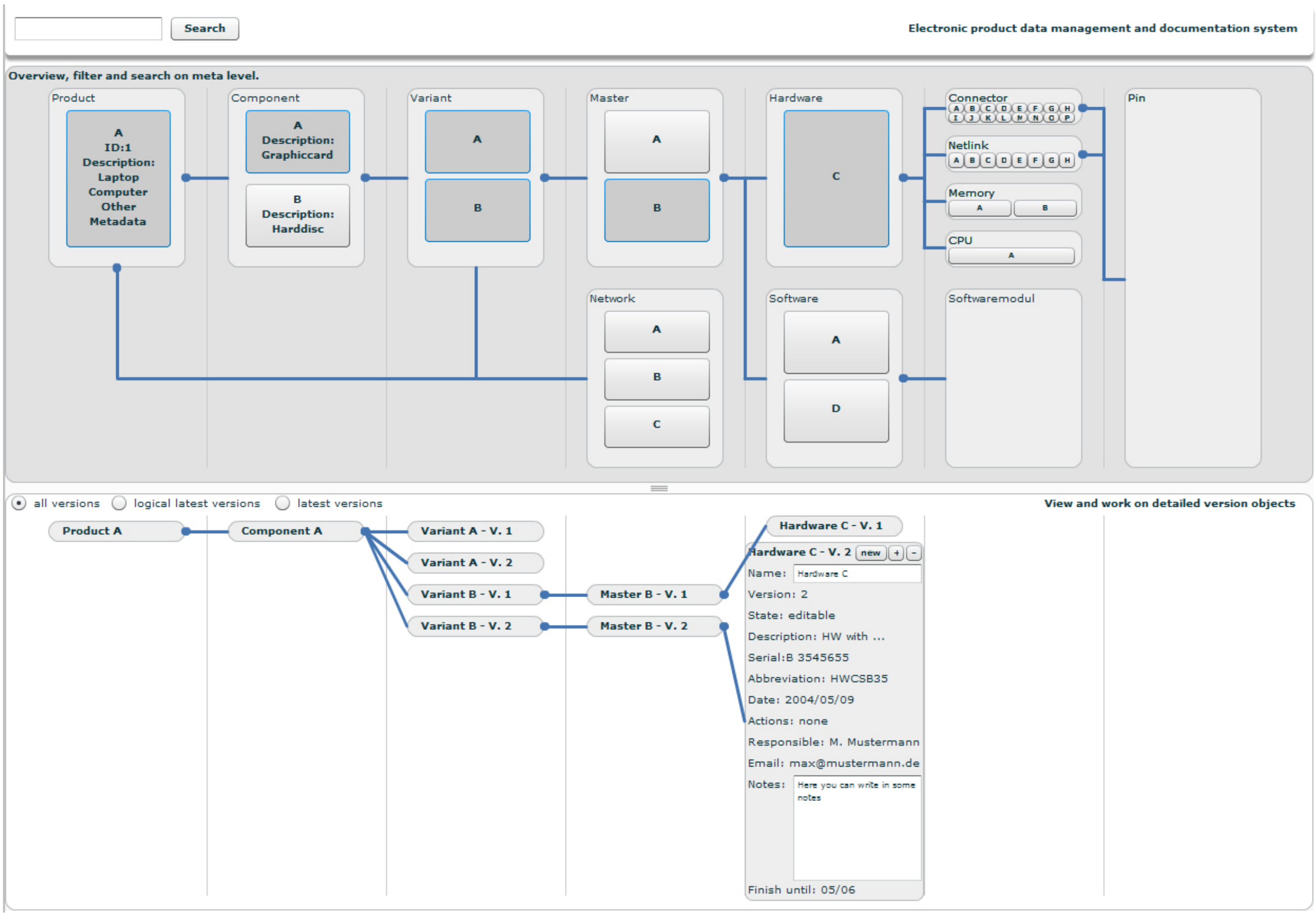

Figure 3: Semantic drill-down into the version object "Hardware C – V.2"

Both the chronological and logical relationships of versions must be displayable at once and it is important to enable access to task-relevant contextual information. At present, navigation (2) is mostly based on an expandable, sometimes redundant, hierarchical tree representation, although data is rather highly networked. Consequently, users loose an overview of product-part relationships. We need to give users a permanent clue as to where they are, where they can go, and where they have already been. Users must be able to browse all relevant product-part dependencies on demand. With regard to search (3), instruments for narrowing the information space with filter options are necessary. A corresponding reduction of complexity eases the EPDM process. After all, there is a need to make use of the many facets of HCI research to deliver an innovative UI.

4. O+D Prototype with Zooming

In multiple prototyping cycles we came to the following design solution for our UI. With regards to the user feedback our system provides a usable search, exploration and task support concerning an exemplary and simplified EPDM information space. Compared to the conventional systems mentioned in section 2 our UI provides much more interactivity and more comprehensive filter possibilities. The EPDM system prototype in Figure 3 is realized with Adobe Flex 2, php and a postgres database as backend.

We can only provide a small number of images here for space reasons; further images and two short videos showing the main ideas are available on our website [11]. The final UI consists of three main areas shown in Figure 3. The first one is a small search bar at the top of the screen (referred to as the search area) that enables a full text search. The second area provides the overview, displaying the static structure of the information space with the hierarchical dependencies of the parts (referred to as the overview). The third area visualizes in detail the version objects of all parts that users are working on (referred to as the detail view). We combine an overview displaying the basic hierarchical structure with a multi-focus filter to solve the problems mentioned in section 3. The result of the user input can be shown with different interactive hierarchical visualizations to offer users an easy way to explore, and navigate through, the narrowed information space.

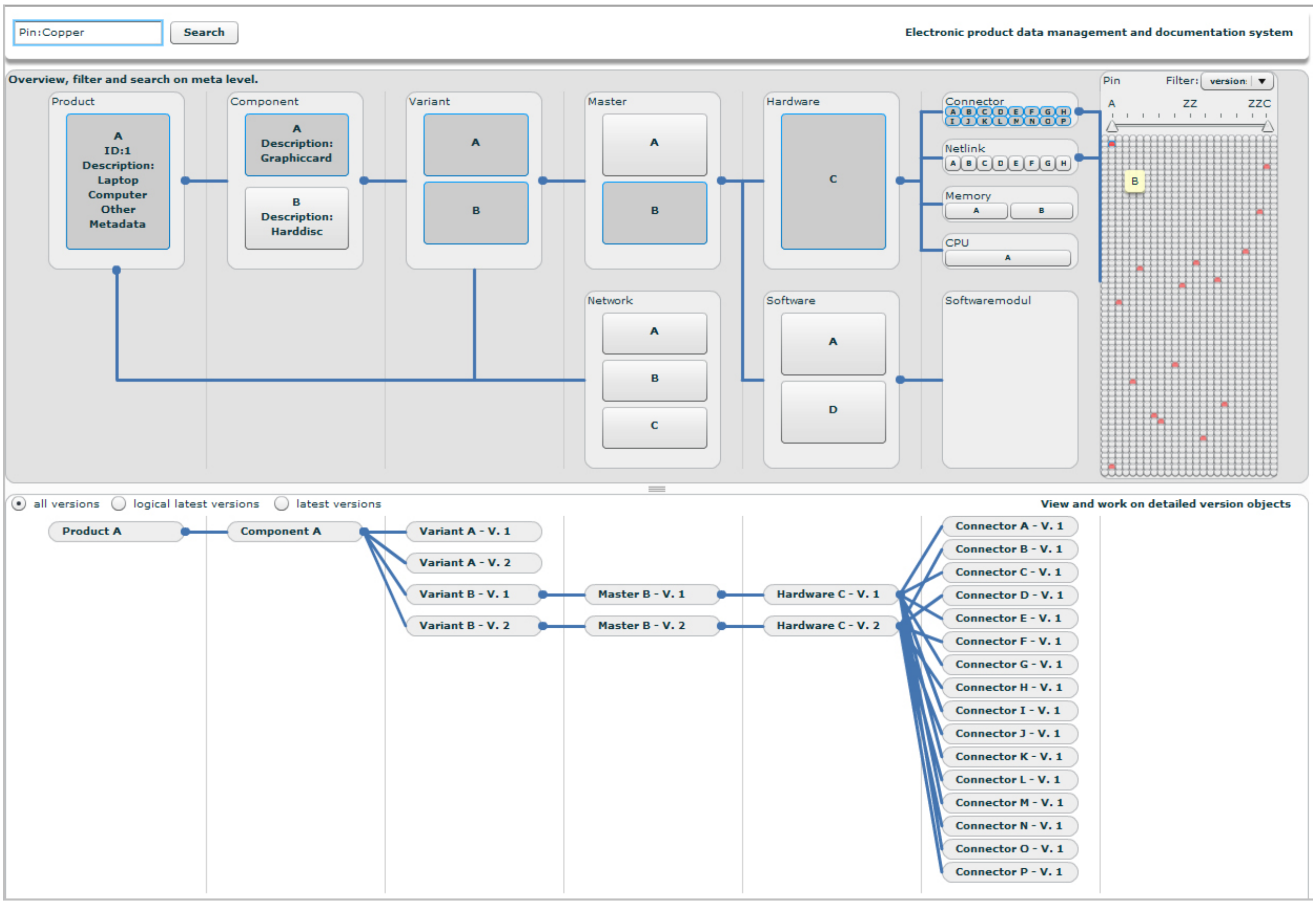

Figure 4: Dynamic query concept and multi-button display

4.1 Visualizing the Information Space

The system visualizes the information space in a multi-view layout, known as the overview and detail (O+D) interface [5]. We use a split screen where both views are of equal size. Most overviews show a miniature of the detail view, however our overview takes the form of an abstract view of the information space. In the overview area, the users can see the different parts of a product, represented as light-grey named areas. We have 13 parts on 7 hierarchy levels in our EPDM data space - from products (left in the overview) at the top level down to pins (right in the overview) at the hierarchically lowest level. Each part group is represented by a toggle button and represents version objects displayed in the same column in the detail view. These version objects are displayed or hidden in the detail view if the related group toggle button is pressed. Selecting and deselecting a toggle button additionally triggers the system to show or hide toggle buttons for lower overview part levels. For example, selecting the "A Description: Graphic card" button in the part *Component* tile reveals the related toggle buttons in the *Variant* tile (see Figure 3). The detail view is changed accordingly by expanding or collapsing the corresponding version nodes. This feature is known as uni-directional tight coupling, or the coordinated view concept [2]. The tight coupling of the O+D interface is additionally highlighted by the grey separating lines that divide the screen vertically according to the quantity of different parts and their hierarchical relationships (see Figure 3, 4).

With the two split-screen views we can provide easy access to the information space. Users maintain an overview of the overall structure of components, while at the same time they can see the version objects and their dependencies. The toggle buttons are the field-of-view [15] in our overview; they indicate which version objects are shown in the detail view (see Figure 3, 4). Research on O+D interfaces shows that users prefer having an overview, and with it the feeling of control, while performing their tasks (see [4], [8] and [9]). The toggle buttons in the overview improve our system in two ways. Initially they act as a kind of intermediate view [15] that is included in the more abstract overview over the whole part hierarchy and its relationships. The toggle buttons provide the user with more information about the available groups of version objects. Our requirements analysis revealed that those groups are of interest to users when narrowing down the information space to match the task situation. We therefore decided to provide users with group toggle buttons as a filter possibility. Hence users can filter the intermediate view itself and the related version nodes in the detail view in an interactive way.

4.2 Browsing the Information Space

The selection of buttons in the overview displays the related master toggle buttons (overview) and the corresponding variant version objects (detail view). This browsing mechanism can be continued and the user can thereby narrow and expand the information space on demand (see Figure 3, 4).
To create a new version or edit an existing one, users can zoom semantically into the object of interest by clicking it in the detail view. In our example, the user zooms into a hardware version. The zoom is highlighted by an animation because users then have a better understanding where they are going to [6]. The individual hardware object enlarges, whereas other objects are pushed aside or may even shrink, dependent on the available screen space. We are therefore now combining the O+D interface with zoomable components for the detail view. Figure 6 shows the different visual representations a version object can have. All the data fields can now be edited and a new version based on the current one can be created. The user can then set up new relationships or adapt existing ones.

4.3 Searching the Information Space

As the information space is very complex, users could display a very large number of objects at any one time. Hence, they must be able to filter and search the information space. In Figure 4 the user fully traversed the parts path down to the connector toggle buttons. This resulted in displaying all related pins in the pin-category tile and all connector version objects in the detail view. As the number of pins can be very extensive (in this case 1500), the system provides various search functions to enable the user to find the product part(s) of interest. Searching for keywords highlights all relevant pin toggle buttons in the corresponding tile of the overview. Users can now easily select the relevant pin buttons and show the related version objects in the detail view. These version objects are additionally highlighted if they contain the relevant search keywords in the user's query.
Furthermore, the system offers a dynamic query solution [2] for each product-part tile. Users can set a filter focus on one or more part tiles in the overview. Clicking on the light grey tile causes it to enlarge and a combo box offers appropriate filter attributes. A slider can be used to narrow down the available number of pin toggle buttons. The more buttons that disappear; the more space can be used to display the other buttons.

4.4 Relate

Relationships between objects can be shown on two levels. The first level is the meta-object level. To show these relationships the overview is used. The blue lines indicate the main connections that exist between the hierarchy levels from product down to pin (see Figure 4). Connections between meta-objects are shown by highlighting. If the mouse pointer is moved over a meta-object toggle button the other related meta-objects are highlighted.

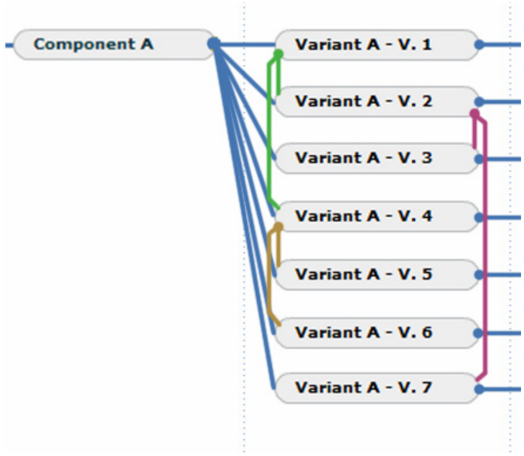

Figure 5: Branching between object versions

Moving the mouse over a node highlights all connected nodes in the detail view. A comparison of two nodes and their relationships is not possible at the moment but in future we want to support multiple node selection. Users then can select more than one node of interest and compare the highlighted connections in both views. Different, easy distinguishable colors should be used to draw the multiple edges. Our first solution to the chronological visualization of object versions is to draw "time"-edges in color different from the hierarchical edges (see Figure 5).

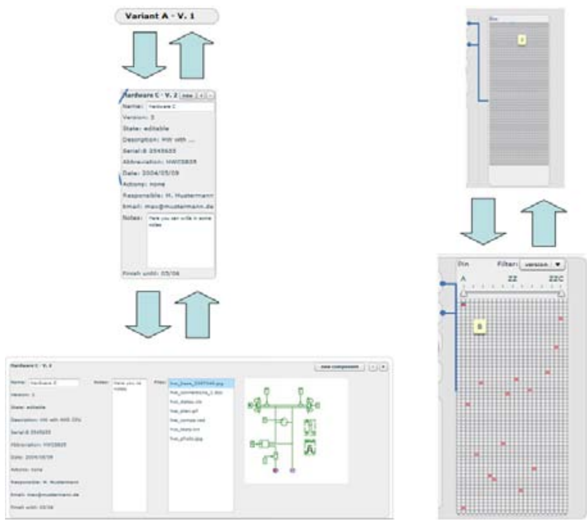

Figure 6: Detail view: zoom of a version object (left), overview: focus of a category area (right).

4.5 Details on Demand

Details on demand are included in our system by the mouse-over highlighting of nodes and relationships in both views. The semantic zoom in (see Figure 6, left) on version nodes to display more information, provides a fast way of navigation between objects. Further a click on a light grey category tile in the overview sets a focus on the tile and expands it revealing more information about the included data (see Figure 6, right).

4.6 Inspiration and Discussion

Although we did not find a system that solved our specific problems, we got some valuable inspiration from publications such as TreePlus [13], where an interactive tree is used to navigate through a graph data space. In the MoireGraph [12], a radial layout is combined with focus and context techniques to show the relevant information to the user without losing the context. We suggest that semantic zooming, distortion techniques like in [3], and pixel displays can be especially helpful for showing relevant information in context. One example is compound fisheye views and treemaps [1]. While an aggregated graph can be expanded by the user, a treemap overview shows the respective hierarchical graph clusters, which are highlighted with the user's navigation steps.

As shown in Figure 4, the height of a column in the detail view may be too small to keep all node labels readable when there are more than about 20 nodes in a column. We plan to solve this problem by improved focus and context techniques, using distortion (as in [17] a fisheye zoom) for each column. This could enable users to maintain context, perhaps even if a node is zoomed to the first level (see Figure 3). The same problem may occur with the overview areas where buttons have to be displayed. In special cases the amount of buttons can be much higher than 1.500. Then we want to use pixel visualizations which can additionally be aggregated if the screen space is still too small.

A major drawback of O+D is often the cost of visual switching. Breaking down information into (window) parts can degrade performance due to the time it takes to mentally relate the views. For instance, [8] compared zoomable UIs with and without overview for map navigation tasks and found that the participants were significantly faster without an overview widget when searching a map with multiple levels (semantic zooming). The decreased performance with the overview widget was at least partially due to the time it took users to visually and actively (by moving the mouse) switch between the two views. We had similar results in our own research [7] and we will therefore develop a detail-only UI based on [17] as an alternative solution. Once finished, we will conduct an evaluation to compare common EPDM tools with our solutions and verify our study with the results from studies such as [5], [8], [9] and [10].

5. Conclusion

We developed an EPDM UI prototype with the Flex 2 web technology, which is new to this domain. The problems we encountered are the amount of data to be visualized, the complex linked data space, the often complicated user tasks and the constraints of intranet applications. Although we admit that we haven't conducted a user study so far, we propose important solutions for the problems mentioned above. The core ideas of our approach are an O+D multiple coordinated views, snap-together visualizations and filters using interactions styles like zooming and dynamic queries on a complex hierarchical structured information space. Semantic zooming and different visualizations are used, dependant on the quantity of displayed information. The information space is enlarged dependant on the focus of interest of the user. Our hypothesis is that the O+D interface provides a much better overview over, and navigation through, the information space than conventional tree navigation EPDM interfaces do. The detail view with its zoomable components enables users to execute their tasks while being aware of where they are, where they have been and where they can navigate to, supported by the visualization of chronological and hierarchical version relationships. As mentioned in 4.6 we will conduct a usability study to prove our hypotheses and get more information on how to optimize our user interface.

Due to the increasing complexity of information spaces in various application domains, advanced UIs are required. In general, an IT organization should consider information visualization, navigation and search when doing UI design. This improves communication, consistency, and lastly the necessary understanding of the overall problem space that has to be made accessible. With our interface concepts we contribute to the solution of visualization, search and filter problems of PLM/EPDM data spaces. The introduction of interactive and adaptive UI takes us a step forward to a fully optimized product lifecycle and the integration and management of all product-relevant information. In future we will try to improve our system to better support a visual analysis of the information space. Dependant on user interaction and user tasks the application must adapt the visualization intelligently and narrow down or expand the information space automatically.

References

[1] Abello J., Kobourov S. G. & Yusufov R. Pach J. (ed.): Visualizing Large Graphs with Compound-Fisheye Views and Treemaps. *Inproceedings Graph Drawing*, New York, 2004, Springer, 2004, pp. 431-441.

[2] Ahlberg C., Shneiderman B.: Visual information seeking: tight coupling of dynamic query filters with

starfield displays. *Proc. of the SIGCHI conference on Human factors in computing systems*, ACMPress, 1994, 313–317.

[3] Bartram L., Ho A., Dill J. & Henigman F.: The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces. UIST '95: *Proceedings of the 8th annual ACM symposium on User interface and software technology*, ACM Press, 1995, 207-215.

[4] Baudisch P., Lee B., Hanna L.: Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In AVI ’04: *Proceedings of the working conference on advanced visual interfaces*, pages 133–140, New York, NY, USA, 2004. ACM Press.

[5] Beard David V., Walker John Q.: Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour and Information Technology*, 9(6):451–466, 1990.

[6] Bederson B. B., Boltman A.: Does Animation Help Users Build Mental Maps of Spatial Information? Tech Report CS-TR-3964, Computer Science Department, University of Maryland, College Park, MD, 1998.

[7] Buering, Thorsten; Gerken, Jens; Reiterer, Harald: Usability of Overview-Supported Zooming on Small Screens with Regard to Individual Differences in Spatial Ability. AVI' 2006: *Proceedings of the 8th International Conference on Advanced Visual Interfaces*, ACM Press, Mai 2006.

[8] Hornbæk K., Bederson B. B., Plaisant C.: Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, vol. 9, no. 4, pp. 362-389, December 2002.

[9] Hornbæk K., Frøkjær E.: Reading patterns and usability in visualizations of electronic documents. *ACM Trans. Comput.-Hum. Interact.*, vol. 10, no. 2, pp. 119-149, June 2003.

[10] Hornbæk K., Hertzum M.: Untangling the usability of fisheye menus. *ACM Trans. Comput.-Hum. Interact.* 14, 2, Article 6 (August 2007), 32 pages, 2007.

[11] Human-Computer Interaction Workgroup, Website to the EPDM topic. URL: http://hci.uni-konstanz.de/epdmod/ (last visited: 30.01.2008).

[12] Jankun-Kelly T.J., Ma Kwan-Liu: MoireGraphs: Radial Focus+Context Visualization and Interaction for Graphs with Visual Nodes. *Proceedings of the IEEE Symposium on Information Visualization*, 2003.

[13] Lee B., Parr C. S., Plaisant C., Bederson B. B., Veksler V. D., Gray W. D. & Kotfila C.: TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, 2006, 12, 1414-1426.

[14] Liu D. T. & Xu X. W.: A review of web-based product data management systems. Comput. Ind. 44, 3 (May. 2001), 251-262.

[15] Plaisant C., Carr D., Shneiderman B.B.: Image-browser taxonomy and guidelines for designers. IEEE Softw., 12(2):21–32, 1995.

[16] Rose Daniel E.: Reconciling information-seeking behavior with search user interfaces for the Web. JASIST 57 (6): 797-799 (2006).

[17] Schaffer D., Zuo Z., Greenberg S., Bartram L., Dill J., Dubs S., Roseman M.: Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Trans.Comput.-Hum.Interact.*, 3(2), 1996, 162–188.

[18] Shneiderman, B. B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proc of the IEEE Symposium on Visual Languages*, IEEE Computer Society Press, Washington, 1996, 336-343.

[19] Stiefel, P. & Müller, J.: A peer to peer based approach to collaborative product engineering. *2nd Conference on "Informations- und Wissensdrehscheibe Produktdatenmanagement"* by "Gesellschaft für Informatik e.V.(GI)", 2006.

[20] Sung, C. S.; Park, Sam Joon: A component-based product data management system. Published online: 30 March 2006, Springer-Verlag London Limited 2006.