Towards Process-Oriented Information Visualization For Supporting Users

Dirk Burkhardt, Tobias Ruppert, Kawa Nazemi
Information Visualization and Visual Analytics
Fraunhofer-Institute for Computer Graphics Research
Darmstadt, Germany
{dirk.burkhardt, tobias.ruppert, kawa.nazemi}@igd.fraunhofer.de

***Abstract*—Nowadays daily office work consists of dealing with big numbers of data and data sources, and furthermore of working with complex computer programs. In consequence many users have problems to use such programs effective and efficient. In particular beginners have significant problems to use the programs correctly due to complex functionality and interaction options. To avoid this overload of the user, the Information Visualization community has recently developed some approaches that aim to support the users. Unfortunately, these approaches are limited to one special aspect, and sometimes they are just appropriate for one special task. Thus, in this paper we introduce a process-oriented user-supporting approach. It allows selecting adequate supporting techniques in correlation to a performed process and activity to guide the user and help him to solve his task. Furthermore, we show the benefits of designing programs and applications, which implement process definitions for the existing tasks to provide the user with better process orientation. This guides the user through difficult and complex processes.**

Keywords: process support, process management, process adaptation, process-oriented information visualization, user-centered interaction, information visualization

I. Introduction

Masses of information and data with heterogeneous attributes and types dominate today's repositories. With the increasing amount of data the access to information gets more and more important.

Different techniques and approaches try to face the problem of information access and acquisition. Methods like information retrieval try to structure the data, whereas structured data already exists in several semantic repositories. Information visualization is one of the most important research areas that develop technologies to extract information from data.

The widespread approach for visualizing search results for queries was commonly applied by list-based representations. This way of representation is the most common way for users and the best known type for representing information in an understandable order. But list-based representations are limited regarding their supported features since they only display ranking. There is no possibility to show relations between two or more data objects. Also hierarchical structures of data are important to be visualized. In the past decades there have been many approaches to use other forms for visualizing information graphically. But these visualizations are limited to specific data-formats.

A key change in the last years of information storing and the possibilities for visualizing information enforced the evolution of the Semantic Web and upcoming new approaches and technologies. The new idea characterizing the Semantic Web is storing information next to the classification of instances into a classification tree in so called triple stores [14]. The approach of storing information in structures or linking nodes with each other is not new, but the combination of it and the definition of triple store data formats like Resource Description Framework (RDF) or Web Ontology Language (OWL) make it powerful and useful for most use cases in research and industry. There also exist some related technologies for semantic data sources. They are using a more flexible format, Linked Open Data (LOD) [15], which contains functionalities similar to those of formalized languages. The semantic annotation of data allows an improved data analysis by computer systems. For example, it allows a semantic analysis by using algorithms for reasoning or inferencing to extract additional information from the existing data.

This basic technological change also allows new approaches for visualizing information. The semantic annotated data makes it possible to create visualizations in dependence of an important aspect [8], not just in relation to the data format. For example, it is possible to create visualization with a special focus on showing the structure of a semantic data source (e.g. relational, geographical or time-depended information). Another advantage is the combinability of different data sources, based on their similarity, which expands the containing features. With the increasing amount of data features, also the possibilities for visualizing aspects of such information increase. With the mass of data and the fast development and introduction of more and better approaches to analyze the data, also the number of analytical tools increases. Today a large number of programs exist for working with data. This is also visible in companies, where most people have to work with computers and specialized programs on big data sources. But this rapid development implies the problem that these programs consist of a high number of integrated functions.

This work has been carried out within the project FUPOL 287119: Future Policy Modeling, partially supported by the European Commission. The FUPOL project proposes a comprehensive and new governance model to support the policy design lifecycle. The innovations are driven by the demand of citizens and political decision makers to support the policy domains in urban regions with appropriate ICT technologies. The foundation of the visualization work is the SemaVis visualization framework developed by the Fraunhofer IGD (http://www.semavis.com).

978-1-4673-2427-4/12/$31.00 ©2012 IEEE

They are designed for working with masses of data and additionally to enable to analyze them too. But these applications are becoming complex and in consequence difficult to use. In summary, the current situation can be seen critical. On the one hand, the programs with their multiple functionalities are required to support an efficient and effective data acquisition and analysis. On the other hand, the user gets more and more overstrained because of the programs complexity, resulting in many users getting lost in data space. The reduction of functionality is often no solution, because that means that less work with the data is possible. Therefore an adequate solution has to get along with no functional reduction. A solution which enables to integrate the full functionality and still helps the user in the usage of the application lies in providing a support strategy based on tasks and processes.

II. VISUALIZATION IN THE INFORMATION GATHERING PROCESS

A. Information Visualization and Semantics Visualization

The common usage scenario of existing search engines is providing search to a concrete result. For instance, if a user is searching for a chancellor's or president's biography, the search engine shows that data entries that fits best to the query [1].

But today's systems allow also other opportunities for searching about less known topics and where no direct result of such a search process is defined. This form of search is named exploratory search. The goal of this kind of search is to gather an overview about a certain topic [2]. For example, if a user heard something about "Cloud Computing" and he does not know what that means, he can perform exploratory search with modern search systems to get an overview. Furthermore a user can gather knowledge he does not expect to get, e.g. if a user is searching for the person "Angela Merkel" as chancellor of Germany, he can also find just namely similar results like the town Merkel in Texas, USA. Hence, exploratory search also allows finding relations to not previously known correlating topics.

For supporting this kind of exploratory search, the user needs appropriate tools. In the last years the research community used semantic visualization systems [3], because of the structured and linked information entities. It enables showing the concrete results and also related information, e.g. sample platforms, articles of newspapers about this topic, etc. A continuative approach is the coupling of additional less or non-semantically annotated data sources [4]. For that a semantic data source is linked with multimedia data-bases [5][6] e.g. YouTube or stream data sources like twitter. This allows a more interactive exploration, where a user gets informed by multimedia elements, which reduces the amount of text to be read. How the information is shown to the users can vary. Some of the existing systems are orienting on special aspects, e.g. SemaPlorer [7], that focuses on visualizing the information on a geographical-map. But there are also multimedia-driven aspects available, where the focus lies on finding pictures, audios or movies to a specific instance. A more user-centered aspect is supported in SemaVis [8] that allows the user to set a focus. Therefore it provides different kinds of visualization, which the user can select and couple to an individualized so called "Knowledge Cockpit". This system allows users to set their own focus on the data and to specify what the user wants to explore for an aspect-of-interest.

B. Visual Analytics

Another key-aspect of the exploration of information is the aggregation of data to provide the user with overviews of the search space. To achieve these overviews the obtainable data has to be preprocessed and analyzed. Due to the large amount of available data the analysis has to be processed with automatic tools (e.g. statistical and data mining tools) for information extraction. The problem that arises here is that the user is confronted with the outcome of the analysis without getting insight in the generation of this output. During the last decade a new research field evolved that addresses the problem of getting access to vast amounts of high-dimensional and complex data. Visual analytics lets users combine their expert knowledge and visual perception capabilities with the storage and processing capacities of computers [34]. Automatic data analysis methods are connected to visualization tools to keep the user in the analysis loop and let her guide the analysis process. This methodology can overcome the described problem for the user of being confronted with results without understanding how this result is achieved.

Following the mantra of [35], visual analytics consists of four stages: analyze first - show the important - zoom, filter and analyze further - details on demand. Inherent in this mantra is the close interaction of the human with the machine.

As a first step in the analysis process, mostly the data has to be preprocessed to bring it into a common structure. Here, the user can interactively set parameters for the automatic preprocessing methods (e.g. outlier detection, missing value detection, etc.). Now the data has to be analyzed to divide the data space into relevant and non-relevant information. Again, the user should be in control of the utilized information extraction methods and the setting of their parameters. The following visualization of the results (interactively selected by the user) gives a first overview of the existing information. Now, the user can decide which information, she wants to analyze further by filtering and zooming. Most important features can be explored by detail on demand. This abstract description of a visual analytics process emphasizes the interaction between user and machine, and underlines the usefulness of data analysis with the support of visualization-enriched user interfaces.

One major challenge of this methodology lies in the complexity of the analysis tasks. At many stages of the analysis process we observe a high degree of freedom regarding the setting of the parameters needed for the automatic algorithms. Here, the user could be guided via a process-oriented design of visual analytics techniques.

C. *Process-Oriented Information Visualization Designs*

All of these systems are highly interactive and support the exploration of an interested topic. But they also include the risk of getting lost in the data. The high degree of freedom in dealing and interacting with such massive data sources is risky, if the user is not exploring the data. So there is a high chance to fail a task, because the mass information does not help to focus on just that part of information, which is really necessary.

Currently just a limited number of existing systems implement a kind of process organization, which e.g. controls the interaction routine between human and applications. But in the past decade some process-oriented approaches have found the way into consumers' applications. Most of them are fixed integrated and there influence character is limited on almost one single aspect. The mostly known implementation of Microsoft Windows users is the installation routine which is a sort of wizard. The installer consists of a process definition. In every step the user gets information or has the ability to take control, e.g. to enter the path where the application should be installed.

Figure 1. A typical installation routine – a sort of wizard of an application in Microsoft Windows.

Furthermore there are a couple of other process-driven approaches which also show a workflow to the user. Mostly shopping websites show such visualizations in the last phase of making the payment for a made order. For this purpose the different steps are shown to the user, e.g. of making the payment, enter the bill address, and the delivery address.

Figure 2. An example of a workflow visualization for the shopping checkout. The image shows the worflow from the website http://www.amazon.co.uk.

All of such systems that are using some of the supporting techniques basing on process-oriented support are providing a significant impact of an improved usability and user experience. Unfortunately the existing implementations are almost limited on just one process-feature. There exists no solution that aims to couple more of them or perhaps in a dynamic way. But it seems beneficial to organize application within processes and enrich it with support features which can be enabled or disabled in dependence of a current step that might be too difficult for a user. For an efficient inclusion of multiple supporting features, there is a need for a classification of existing process-supporting features. As a main benefit applications can be seen, which allow users a more usable and efficient interaction, especially for difficult tasks or in situation where user have to deal with various data sources.

III. General Approaches for Supporting Users within Visualizations

To avoid the risk that a user explores passing the aimed goal, approaches for supporting users through the interaction process within an application exist. To allow a better overview we categorize the approaches into different groups, as a kind of survey. They all have as goal to support the user to solve given tasks more efficient, effective, and in a user-satisfying way.

A. *Adaptation*

Adaptation is mostly implemented by considering context variables, like the interaction history of the user or the different ranking of data entities. Based on that, the graphical interface is automatically changed to improve the representation adapted to the user's behavior. The way of adaption can vary, so it is possible to change just some visual parameters, e.g. the size, color or the position on the screen, or perhaps the entire visualization through changing the layout or replacing the coupled visualizations.

Another possibility for supporting users in relation to the processes is using personalization and adaptation approaches. Focusing on semantic technologies, these approaches can be subdivided into three types [9]:

1) The first is adaptation of content and recommendation. It aims at highlighting entities which fit best to the user, e.g. based on the deviation of the entity names in perspective to a user's performed search query.

2) The second option is adaptation of structure and presentation, which encompasses the customization of the data structure in a user menu. Therefore the hierarchy of the structure can be changed by sorting, hiding, visualizing, enabling and disabling the existing concepts.

3) The third approach is adaptation to support annotation, which is often realized by tagging and the user get the possibility to get content tagged by other users by his own often used tags.

B. *Recommendation*

Another aspect is recommendation. Recommendation can also be included in adaptive systems. But in general recommendation is the ability to advice the user for a possible improvement of the visualized data or toolset. Recommendation is mostly based on interaction data of a group of users, or on the underlying content. So, normally the people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients [10].

In the first developed recommender system from Tapestry, this approach was named "collaborative filtering", which is also named in some recent publications [11]. But this term is confusing, because some approaches do not explicitly collaborate with recipients [10].

To differ the existing recommendation systems Gaul et al. defined a three-dimensional classification regarding the input-data [13][12]:

1) Explicit versus implicit input data. The explicit approach based on direct user feedback, which comes from a directly asking the user for their opinion on objects. In contrast, implicit recommendation is based on behavioral usage data.

2) Degree of personalization. To allow the presentation of recommendation to appropriate users it is necessary to identify these users first. The identification can be performed on different levels, beginning from low, e.g. transaction or session, level up to the highest level, e.g. full user identification and search context.

3) User-centered vs. item-centered. The user-centered approach focusses on the interests of the users. Based on the interests the users are grouped.Later the recommendation is performed on these interests for the existing different groups. This method is less anonym. In contrast the item-centered approach neglects the user and his interests. The recommendation is just performed on the items and its character . E.g. on shopping websites only similar products will be shown.

C. Simplification in User-Interface Design and Functions on Demand

The simplest way to support users is to provide an easy understandable user-interface, which just offers the main required functions. Such approaches can be grouped into "simplification and functions on demand". The main idea is to reduce the functions to the most significant and requested ones - this also counts for the conceptual model [16]. Next to the visualization and general application design, a simple and retraceable interaction design is helping the user to deal with a system [18]. The main challenge for these approaches is to determine those functions that fit well to the users' goals and tasks. Therefore the applications need to be efficient, effective, and needs a well users' satisfaction. To measure the usability of user-interfaces the characteristics efficiency, effectiveness and user's satisfaction are the main observed values [17].

An advanced approach of simplification lies in limiting only the visible functions. Therefore the simplification takes just place on the visible user-interface, where just those function can be selected, that are useful for the current display. In consequence, the list of functions depends on the displayed situation e.g. if a performed search has no data, then also no data-editing function should be displayed. It is also possible to create stereo-type user-interfaces. Common types are easy and advanced modes. The easy mode is designed for application novices, only fundamental functions are provided. The advanced mode is mostly designed for experts. A bigger set of functions is provided. This simplification approach is summarized as *functions-on-demand*.

D. Orientation, Guidance, Wizards, and Workflows

Mostly, recently published software becomes larger and more complex. It can be seen as a trend to provide as much functionality to users as possible. This results in difficult to understand applications and user-interfaces where most users do not understand how to use them. To solve this issue, the incorporation of explicit user guidance into toolsets is an effective opportunity, which can be named as user guidance environment [19]. A mostly used approach for guiding the user is a hidden defined workflow in an application, on which the communication of a user can be allocated to a specific step within the workflow. The advantage of such an implemented workflow is that the process is mainly regarded as a whole [21]. The workflow can contain a single path to do one task. In some cases it can also contain alternative ways or perhaps ways of applying functionality to solve this task [20][21]. In contrast to such general workflows, the guidance can also be implemented with Wizards. Wizards are beneficial if users should be guided through a fixed process e.g. installation routine (as presented in Figure 1). Within the different steps the users get informed or have to enter required information [23].

A continuous approach of implementing a workflow in applications is the visualization of such a workflow. The main idea is to provide orientation for the user so that she sees in which phase of a process she is acting. The visualization can vary between just a numerical value e.g. "you are on page 5 of 10" or in graphical form like they are used on some web shops and pictured in Figure 2.

E. Hint and Highlight

To support users during the work with an application, the system can help users by showing hints e.g. tool tips or emergency warnings. These approaches do present information in the user's view to help him to solve his task faster. As an advantage the user does not need to search a special function, instead the system advices the user to the special function [25]. The view of the user is directed to the information that might be important for him. The existing hints can be statically designed or they can be automatically generated [24].

A similar approach is the highlighting of information. Here the goal is to guide the user's attention to the visualized information that might be interesting for a user. So, the user sees all the information, but expected information with a higher importance to her is marked, e.g. with a different background color [26][27].

Both approaches are primary designed to set the focus of the user on specific information. There are no processes needed, but such functions are often used in process-driven scenarios.

F. Key Strokes and Shortcuts

Under this approach we summarize all alternative interaction forms with the keyboard to allow the interaction in graphical applications that normally have to be controlled with the mouse. In general we can distinguish two types of keyboard interactions: keyboard shortcuts (or hotkeys) and stroke shortcuts (or stroke gestures) [28]. The benefit is the high efficiency, because instead of clicking with the mouse through a context menu for a specific command, just a simple shortcut on the keyboard has to be pressed to execute the same command. This allows a quick and easy interaction.

This kind of support is predominantly used by advanced users, because it is necessary to know about that feature in general. Moreover, the existing keyboard shortcuts and stroke shortcuts of the operating system and within the application have to be known. Another problem is the missing support in modern devices, for instance on tablets and smartphones, which do not provide a keyboard as common interaction interface.

G. Alternative Interaction Methods

The interaction-process for users can be designed easier if alternative interaction devices will be used with more natural interaction strategies. Especially interaction forms which are close to natural human interaction, e.g. with the performance of gestures [31], can make the communication with technical systems easier. Therefore the application has to support such devices, or a system that takes on the organization of such devices is needed [30][29]. The communication by speech also counts to alternative interaction forms.

IV. Approach for Supporting Users with Process-Oriented Visualizations

A. General Terms for Process-Oriented User Support

The terms *process*, *process management* and *activity* which we use in this paper have the same meaning as defined in [32][33]. Regarding the goal of supporting users through processes, we have to define the terms process support and activity support. The goal of process support is to help the user to complete a task by regarding her maxim, e.g. to design the process efficient, effective, easier, traceable etc. To achieve that, the support is oriented on the process, its sub-processes and activities. Under this background we define process support generally as follows:

The use of Process Support aims to improve the qualitative or quantitative result as outcome within a performed process by a user. Therefore features are additionally integrated to ensure that the process can be performed in a better way.

In scope of providing process support within a technical system, this definition can be concretized as follows:

The use of Process Support in a technical environment aims to improve qualitative or quantitative results as outcome within a performed process by a user through consideration of process-specific aspects. Therefore technical features have to be considered or implemented which ensure that the process can be perform in a better way.

A step which requires the interaction of the user is an activity. Close to the definition of process support, we have to define activity support, too. We define activity support as follows:

The use of Activity Support aims to improve the qualitative or quantitative result as outcome within a performed activity as piece of work that forms a logical step within a process. Therefore features are additionally integrated for a single activity or the entire process to ensure that the activity can be performed in a better way.

The focus of the definition of these terms was to characterize their main character and goal. It contains no limitation or information about a possible technical implementation. The advantage of this abstract definition is the ability to classify also existing technical solution from research and industry as process supporting technology.

B. Explicit and Implicit Definition of Process-Orientation

The implementation of process-orientation can be applied explicitly or implicitly. The most approaches are implicit implementations, so the process is not precisely defined. Even more they are indirectly defined. For example when a user can select a setting menu-item in graphical application, she gets a form to change them, and at least if the user clicks on save, the changed settings are stored. The process which is described in this example is (1) select the setting menu-item, (2) change the properties, and (3) store and apply the changes. This is just a simple process. There are a couple of other processes (represented through the other menu-items). The indirect processes are often difficult to characterize by their single activities, because next to direct user interaction there are also some technical internal operations that can be necessary to perform the whole process.

Explicit process-oriented implementations are using a process model with concrete process definition, e.g. in BPEL [40] or XPDL [41]. The main advantage is the possibility of modeling the process and activity in detail just by definition. This allows the adoption of processes to changed environments, because among others, validators and performed functions can be set and changed without investing additional implementation effort.

C. Integration of Process Support Features

The integration of concrete process support features depends on the implementation of process-orientation. If process-orientation is implemented implicitly it has to be integrated directly into the source-code of an application. This means that just the developers are able to define the concepts how process-relevant data is collected, analyzed it, and how the results are affecting the visualized data, e.g. via adaptation or recommendation.

If the process-orientation is implemented explicitly, the integration of a process support feature can be easier. In some cases such features can simply integrated through the process-definition. Also changes on the parameters for the system can be changed easier.

D. Concept for User-Supporting Process-Driven Information Visualization

The concept distinguishes the control of the processes and the control of generating visualizations. Still, there exists a communication between them (see Figure 3). The main part for the visualization generation is the visualization control, which contains all issues to generate the graphical elements for the final view. The main components needed for a visualization is (a) a data connector, which accesses the available data from a data source. It can also include data transformation functionality, to make the data compliant for the following component. The visualization controller (b) is the second needed component. It organizes the mapping of the incoming data from the data connector to the visualization (c). To allow an adequate processing it is recommended to implement a processing pipeline to ensure an optimal visualization of the data. The most established is the reference model for visualization from Card et al. [38], but there are also pipelines for supporting special types of visualizations and data e.g. the Semantics Visualization pipeline by Nazemi et al. [39].

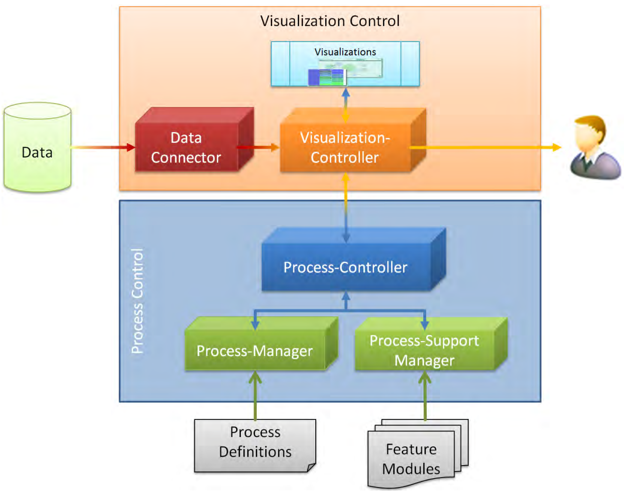

Figure 3. General concept for an process-oriented Information Visualization system that supports integration of process-supporting features.

The process support is than part of the process control. It consists of two parts. The process manager is responsible for the organization of the existing processes. The processes need to be defined first, but they can also be integrated into a learning system, which empowers the process generation during use. All processes, the linking between them, and furthermore the activities of the processes need to be organized in this module. The second part of the process control is the process-support manager, which organizes all implemented process supporting features (an overview to the general techniques was given in chapter III). To support the user, the process-controller collects context information, e.g. the current process and activity. Based on that information the controller decides which process supporting type is appropriate for the current process, respectively the current task.

E. Application in Information Visualization Systems

In the past we have developed a semantic visualization framework that provides a number of approaches to allow the user to perform an effective semantic-based search and exploration. In the past we used implicit process-oriented search and exploration strategies which have limited the opportunities to integrate supporting features, e.g. for an adaption module. We have created a module to adapt the visualization based on the user's interaction [39]. Because of the missing concrete process definition, the integration had to be implemented directly in the source code and was adapted to the framework-based technical environment.

In an updated framework design, we have integrated the presented process-oriented concept with an explicit process-oriented definition, which now allows the definition of various processes over a BPEL configuration file. The supporting features can now be developed as modules, and with the configuration it is possible to define how this support feature will be integrated. In a first implementation we have just implemented a single process supporting feature, a user-interaction analysis module. This module analysis the interaction history of a user and highlights frequently selected entities. In dependency of the resulting amount of entities from a search query, the developed process supporting system enables and disables automatically the user interaction analysis module. The configuration since which amount of entities the module is enabled or disabled, and also the concrete adaption strategy (how to highlight an entity – e.g. by color or size) was just defined via the BPEL configuration file. An example screenshot of the system is depicted in Figure 4.

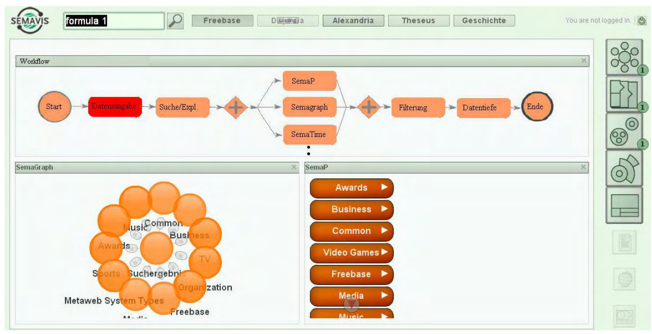

Figure 4. A semantic visualization system that uses the general concept for an process-oriented information visualization system.

V. CONCLUSION

The paper has summarized the current situation, that on the one hand data sources are increasing, and on the other hand the complexity of applications also increases. This often has a negative influence on the satisfaction of the users, who have to use the application and the data. Next to the satisfaction there is also the challenge, that users have problems in understanding the application. The interfaces are confusing the users, and this in fact decreases the productivity. To avoid overstraining the user, the community has developed several approaches to support users in the application usage. Unfortunately, most applications are implementing just one of these approaches. Furthermore, not every supporting approach is appropriate for every task. Hence, to allow a better selection of process supporting features, it is essential to have knowledge about the process.

In this paper we introduced a novel approach to determine adequate process supporting features, relating to a current process or activity. We have classified the existing solutions for process supporting features first. After that, we have defined several terms for process support, and we have described in detail a general concept for the implementation of a process-oriented user support. This general concept enables to determine, what type of process support is optimal for a defined process within an application.

VI. DISCUSSION

The presented approach is just oriented on the process itself. For many scenarios it supports the users in solving their tasks in a more satisfying way. Hence, there are also scenarios, where just the information about the process is not appropriate enough, because there are user-related aspects that are neglected. For instance, if a user is color-blind, and the used process-support for a graphical visualization is adaptation by color - it does not has any affect, or perhaps a negative effect, because the color of an important red-marked node looks brighter for him and thus unimportant, then a green-marked node. Another example is the use of hints. If an expert is acting with an application and he gets hints that are primary designed for beginners, the support is inadequate. So there is a need to consider also the expert-level of a user. In general, besides incorporating knowledge about the process, it seems also interesting to collect knowledge about the user and conceptualize a model, which considers the user's behavior in addition to the process knowledge.

REFERENCES

[1] M. L. Wilson, "Search User Interface Design," Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgen & Claypool Publishers. 2011.

[2] R. W. White, S. M. Drucker, G. Marchionini, M. Hearst, and M. C. Schraefel, "Exploratory search and HCI: designing and evaluating interfaces to support exploratory search interaction," In CHI '07 extended abstracts on Human factors in computing systems (CHI EA '07). ACM, New York, NY, USA, pp.2877-2880, 2007.

[3] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou, "Ontology visualization methods - a survey," ACM Comput. Surv. 39, 4, Article 10, 2007.

[4] A. Gupta, K. K. Viswanathan, A. Joshi, T. Finin, and P. Kumaraguru, "Integrating linked open data with unstructured text for intelligence gathering tasks," In Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011 (IIWeb '11). ACM, New York, NY, USA, 2011.

[5] A. J. Roa-Valverde, "Multimedia information retrieval as a practical application for interlinking approaches," In Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11), Chiara Ghidini, Axel-Cyrille Ngonga Ngomo, Stefanie Lindstaedt, and Tassilo Pellegrini (Eds.). ACM, New York, NY, USA, pp.230-233, 2011.

[6] R. Troncy, B. Malocha, and A. T. S. Fialho, "Linking events with media," In Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS '10), ACM, New York, NY, USA, 2010.

[7] S. Schenk, C. Saathoff, S. Staab, and A. Scherp, "SemaPlorer-Interactive semantic exploration of data and media based on a federated cloud infrastructure," In Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Volume 7, Issue 4, pp.298-304, 2009.

[8] K. Nazemi, D. Burkhardt, M. Breyer, C. Stab, and D. W. Fellner, "Semantic Visualization Cockpit: Adaptable Composition of Semantics-Visualization Techniques for Knowledge-Exploration," In Proceedings of International Conference Interactive Computer Aided Learning (ICL 2010), University Press, Kassel, pp.163-173, 20120.

[9] I. Torre, "Adaptive systems in the era of the semantic and social web, a survey," In Journal of User Modeling and User-Adapted Interaction, Vol. 19, Is. 5, pp.433-486, Springer, Netherlands,2009.

[10] P. Resnick, and H. R. Varian, "Recommender systems," In Magazine of Commununications of the ACM, Vol. 40, Iss. 3, ACM, New York, NY, USA, pp.56-58, 1997.

[11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," In Magazine of the ACM, Vol. 35, Iss. 12, ACM, New York, NY, USA, pp.61-70, 1992.

[12] A. W. Neumann, "Recommender Systems for Scientic and Technical Information Providers," Diss., University at Karlsruhe (TH), 2008.

[13] W. Gaul, A. Geyer-Schulz, M. Hahsler, and L. Schmidt-Thieme, „eMarketing mittels Recommendersystemen," In Marketing–Zeitschrift für Forschung und Praxis (ZFP), pp.47–55, 2002.

[14] J. Cardoso, „The Semantic Web: Real-World Applications from Industry", Springer US, 2007.

[15] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data – The Story So Far," In International Journal on Semantic Web and Information Systems, 5(3), pp.1–22, 2009.

[16] J. Johnson, "Designing with the Mind in Mind: Simple Guide to understanding User Interface Design Rules," Morgan Kaufmann, Burlington, MA, USA, pp.136-141, 2010.

[17] J. Nielsen, "Usability Engineering," Morgan Kaufmann, pp.165-206, 1994.

[18] D. Wigdor, and D. Wixon, "Brave NUI World: Designing Natural User Interfaces for Touch and Gesture," Morgan Kaufmann, pp.23-26, 2011.

[19] T. J. Sliski, M. P. Billmers, L. A. Clarke, and L. J. Osterweil, "An architecture for flexible, evolvable process-driven user-guidance environments," In SIGSOFT Softw. Eng. Notes Vol. 26, Iss. 5, pp.33-43, 2001.

[20] K. Pohl, K. Weidenhaupt, R. Dömges, P. Haumer, M. Jarke, and R. Klamma, "PRIME - toward process-integrated modeling environments," In 1. ACM Trans. Softw. Eng. Methodol. Vol. 8, Iss. 4, pp. 343-410, 1999.

[21] C. Günther, S. Schönig, and S. Jablonski, "Dynamic guidance enhancement in workflow management systems," In Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12). ACM, New York, NY, USA, pp. 1717-1719, 2012.

[22] L. Bergman, V. Castelli, T. Lau, and D. Oblinger, "DocWizards: a system for authoring follow-me documentation wizards," In Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST '05). ACM, New York, NY, USA, pp. 191-200, 2005.

[23] D. Silingas, S. Pavalkis, A. Morkevicius, "MD wizard - a model-driven framework for wizard-based modeling guidance in UML tools," In Proceedings of International Multiconference on Computer Science and Information Technology, IMCSIT '09, pp. 609-615, 2009.

[24] E. B. Charrada, and M. Glinz, “An automated hint generation approach for supporting the evolution of requirements specifications,” In Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE-EVOL '10). ACM, New York, NY, USA, pp. 58-62, 2010.

[25] K. Hjert-Bernardi, J. Melero, D. Hern'ndez-Leo, "Comparing the Effects on Students' Behavior of Two Hint Techniques Embedded in a Digital Game-Based Learning Tool," In Proceedings of IEEE 12th International Conference on Advanced Learning Technologies (ICALT), pp.138-140, 2012

[26] M. Ostkamp, G. Bauer, and C. Kray, “Visual highlighting on public displays,” In Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis '12). ACM, New York, NY, USA, Article 2 , 6 pages, 2012.

[27] E. Lin, S. Greenberg, E. Trotter, D. Ma, and J. Aycock, “Does domain highlighting help people identify phishing sites?”, In Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, pp. 2075-2084, 2011.

[28] C. Appert, and S. Zhai, “Using strokes as command shortcuts: cognitive benefits and toolkit support,” In Proceedings of the 27th international conference on Human factors in computing systems (CHI '09). ACM, New York, NY, USA, pp. 2289-2298, 2009.

[29] D. Burkhardt, M. Breyer, C. Glaser, C. Stab, K. Nazemi, A. Kuijper, “Classifying Interaction Methods to Support Intuitive Interaction Devices for Creating User-Centered-Systems,” In Universal Access in Human-Computer Interaction: Part I : Design for All and eInclusion. Lecture Notes in Computer Science (LNCS) 6765, Berlin, Heidelberg, New York: Springer, pp. 20-29, 2011.

[30] D. Burkhardt, C. Stab, M. Steiger, M. Breyer, K. Nazemi, „Interactive Exploration System: A User-Centered Interaction Approach in Semantics Visualizations, “ In Proceedings of Cyberworlds, Darmstadt, Germany, 2012.

[31] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, “Gesture recognition with a Wii controller,” In Proceedings of the 2nd international conference on Tangible and embedded interaction (TEI '08). ACM, New York, NY, USA, pp. 11-14, 2008.

[32] Workflow Management Coalition. http://www.wfmc.org/. [Online - accessed September 2010].

[33] Workflow Management Coalition Terminology & Glossary, February 1999.

[34] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, “Mastering The Information Age - Solving Problems with Visual Analytics”, Eurographics, 2010.

[35] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, 2008. Visual Analytics: Scope and Challenges. In: Visual Data Mining, S. J. Simoff, M. H. Böhlen, and A. Mazeika (Eds.). Lecture Notes In Computer Science, Vol. 4404, Springer-Verlag, Berlin, Heidelberg, pp. 76-90, 2008.

[36] P. C. Wong; an J. Thomas, "Visual Analytics," Computer Graphics and Applications, IEEE , vol.24, no.5, pp. 20- 21, 2004.

[37] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and Guy Melancon, “Visual Analytics: Definition, Process, and Challenges,” In Information Visualization, Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North (Eds.). Lecture Notes In Computer Science, Vol. 4950. Springer-Verlag, Berlin, Heidelberg, 154-175, 2008.

[38] S. K. Card, J. D. Mackinlay, B. Shneiderman, “Readings in Information Visualization: Using Vision to Think,” Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[39] K. Nazemi, C. Stab, A. Kuijper, “A Reference Model for Adaptive Visualization Systems. HCII'11 Proceedings of the 14th international conference on Human-computer interaction: design and development approaches, pp. 480-489, 2011.

[40] F. Leymann, u.a.: Business Process Execution Language for Web Services.– ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf, 2003.

[41] Workflow Management Coalition, “Process Definition Interface-- XML Process Definition Language,” http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20%282012-08-30%29.pdf, 2012.