Chapter 9
Visual Process Support to Assist Users in Policy Making

Dirk Burkhardt
Fraunhofer Institute for Computer Graphics Research (IGD), Germany

Kawa Nazemi
Fraunhofer Institute for Computer Graphics Research (IGD), Germany

Jörn Kohlhammer
Fraunhofer Institute for Computer Graphics Research (IGD), Germany

ABSTRACT

The policy making process requires the involvement of various stakeholders, who bring in very heterogeneous experiences and skills concerning the policymaking domain, as well as experiences of ICT solutions. Current solutions are primarily designed to provide "one-solution-fits-all" answers, which in most cases fail the needs of all stakeholders. In this chapter, the authors introduce a new approach to assist users based on their tasks. Therefore, the system observes the interaction of the user and recognizes the current phase of the policymaking process and the profile of the user to assist him more sufficiently in solving his task. For this purpose, the system automatically enables or disables supporting features such as visualization, tools, and supporting techniques.

INTRODUCTION

Stakeholders often have to consider big datasources during policy making, be it statistic data, social media data or something else. In most cases the stakeholder has to deal with complex systems, which allow on the one hand extensive analysis, but on the other hand require some amount of expert knowledge to use them correctly and effectively. Therefore, most visual information processing systems are balanced between little functionality but simple use, majorly focusing on experienced, but not expert users, and a high amount of functionality with high complexity which are therefore mostly designed for experts.

The major challenge between these two options, simple systems for ordinary users and complex systems for experts, is that especially if systems

DOI: 10.4018/978-1-4666-6236-0.ch009

are designed for a larger target user group, not all users are represented adequately. One alternative is to provide different systems (or different views and tools within the system for the different stereotypes of users) depending on the users' expertise. However these systems are mostly not very efficient for all users. Usually the user group in the center of the spectrum between expert and novice users is in the uncomfortable situation of having to select one of the non-optimal two solutions. Additionally, this user group has to deal with the context change, if they switch between these two stereotype views, which come along with a hard break in the provided visual and interaction metaphors. Such stereotype programs are known also from daily use, e.g. some CD burning programs provide users a simple burning mode and, an advanced one. Both modes do look different and provide a different feature list.

Based on many of these kinds of experiences the user can be identified as the main actor, whose behavior should be the top priority during the concept and design phase. Considering the user-based aspects more sufficiently in computer systems, the development of user-centered systems or user-centered information visualization became more and more important. In contrast to other approaches the development is not only driven by the possibilities (e.g. what can be done with the data?), but also and even more important is that the user will be able to handle the system and solve his tasks. In order to better address the behaviors of the user, various strategies were developed to support him in solving the tasks. The solutions vary in their strictness, solutions like the implementation of shortcuts are useful features that might be used, but they can also be ignored. Other supporting solutions such as wizards are very strict, so the user has to follow without any option to break out of this routine. On the one hand such a strict user limitation can be annoying, if the user knows what to do and how to do it. That can result in situations where experts have to solve a task in a way that is not very efficient. On the other hand, such a strict user limitation often leads to a successful task solving, whereas additional useable features can be ineffective and therefore they imply the risk of failing.

In this chapter we describe a novel adaptive approach to support and assist the user during his task mainly based on the actual phase in the process, but even more based on the user's behavior. For certain tasks and interactions of the user, different technical features that support and assist the user can be enabled or disabled. Such technical features can be, for example, visualization, which shows a specific issue more precisely. Furthermore, technical features can be tools, such as an editor or supporting techniques, e.g. recommender techniques or hint techniques. In fact, the user gets no stereotypic changing views, instead he gets additional features enabled or disabled in dependence of his work and his personal needs, so that the user does not lose track of his goal, just because of a clear understandable user-interface. In consequence, experts will get less restrictive features and tools that allow solving tasks more effective, whereas novices will mostly get restrictive features where they are more strictly guided through the task solving routine and get features and visualization recommended that i.e. other novices used to solve the task.

CLASSIFICATION OF USER-ASSISTANT TECHNIQUES

To avoid the risk that a user explores passing the aimed goal, approaches for supporting users through the interaction process within an application do 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 in solving a given task more efficiently, effectively, and in a user-satisfying way.

In general, we can classify the general approaches into seven distinct categories (see Figure 1). The difference can more precisely differ how static or dynamic they are. Static features are approaches that are almost always available to the user in the same form. The simplest example for such a static approach is key strokes, e.g. Ctrl+C or Ctrl+V for copying and pasting data, whereas dynamic features are just available under certain conditions, e.g. adaptations of the user-interface based on probabilistic model.

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.

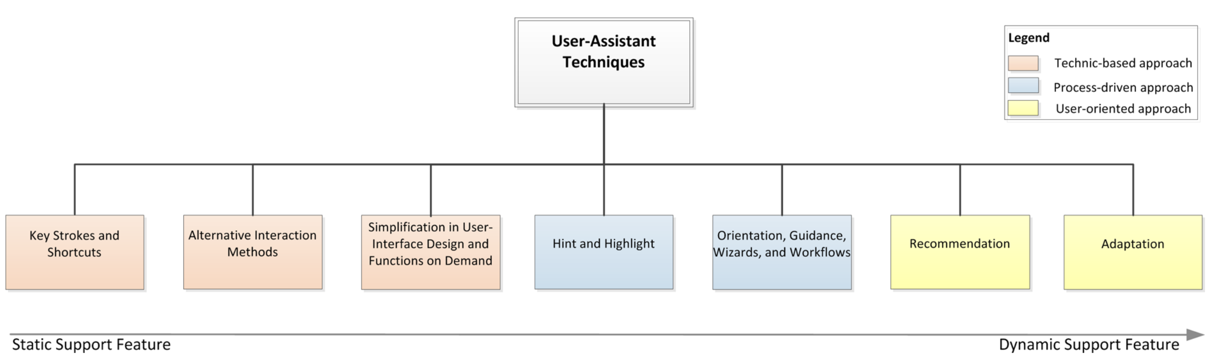
Another possibility for supporting users in relation to the processes is using personalization and adaptation approaches. Focusing on semantic technologies, these approaches can be subdivided into three types (Torre, 2009):

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 with respect to the 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 about adaptation to support annotation, which is often realized by tagging. The user also gets the possibility to get content tagged by other users and based on his own, often used tags.

Recommendation

Another aspect is recommendation. Recommendation can also be included in adaptive systems. But

Figure 1. The classification taxonomy of user assistent techniques. The position from left to right represents its level of dynamics. A static feature ignores the personal needs of user and is available all the time, whereas dynamic features strongly orienting on the user needs, and providing individualized support based on these needs.



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. Normally people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients (Resnick & Varian, 1997).

In the first developed recommender system from Tapestry, this approach was named "collaborative filtering", which is also the name in some recent publications (Goldberg et al., 1992). However, this term is confusing, because some approaches do not explicitly collaborate with recipients (Resnick & Varian, 1997).

To differentiate the existing recommendation systems, Gaul et al. defined a three-dimensional classification regarding the input-data (Gaul et al., 2002; Neumann, 2008):

1. Explicit versus implicit input data. The explicit approach based on direct user feedback, which comes directly from asking the user about their opinion on different subjects. In contrast, implicit recommendation is based on behavioral usage data.
2. Degree of personalization. To allow for the presentation of recommendation to appropriate users, it is necessary to first identify these users. The identification can be performed on different levels, beginning from low, e.g. transaction or session, up to the highest level, e.g. full user identification and search context.
3. User-centered versus item-centered. The user-centered approach focusses on the interests of the users. Users are grouped based on the interests. Later, the recommendation is performed on these interests for the existing different groups. This method is less anonymous. 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.

Simplification in User-Interface Design and Functions on Demand

The simplest way to support users is to provide an easy-to-understand user-interface, which offers just the main required functionality. 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 (Johnson, 2010). Next to the visualization and general application design, a simple and retraceable interaction design is helping the user to deal with a system (Wigdor & Wixon, 2011). 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 high level of user satisfaction. To measure the usability of user-interfaces, the characteristics efficiency, effectiveness and user's satisfaction are the main observed values (Nielsen, 1994).

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.

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 (Sliski et al., 2001). An often 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 (Günther, Schönig & Jablonski, 2012). 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 (Pohl et al., 1999; Günther, Schönig & Jablonski, 2012). 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 (Silingas et al., 2009).

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 he sees in which phase of a process he is currently 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.

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 (Hjert-Bernardi, Melero & Hern'ndez-Leo, 2012). 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 (Charrada & Glinz, 2010).

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. Thus, the user sees all the information, but expected information with a higher importance is indicated, e.g. with a different background color (Ostkamp, Bauer & Kray, 2012; Lin et al., 2011).

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.

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 performed using the mouse. In general, we can distinguish two types of keyboard interactions: keyboard shortcuts (or hotkeys) and stroke shortcuts (or stroke gestures) (Appert & Zhai, 2009). The benefit is the high efficiency, because instead of clicking with the mouse through a context menu and searching for a specific command, only 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 remembered. 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.

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 (Schlömer et al., 2008), 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 (Burkhardt, Stab, Steiger, Breyer & Nazemi, 2012; Burkhardt et al., 2011). The communication by speech also counts to alternative interaction forms.

DESIGN OF A PROCESS-BASED USER SUPPORTING SYSTEM

In the previous chapter we introduced the current state of the art approaches to support users. In this chapter we outline our novel process-adaption system to support and assist users. The major idea is to observe all internal system events, as well as user interactions with the system. Based on these events, a generalized process model will be generated in the first phase. In the second phase, this process model supports the user through enabling and disabling of various features.

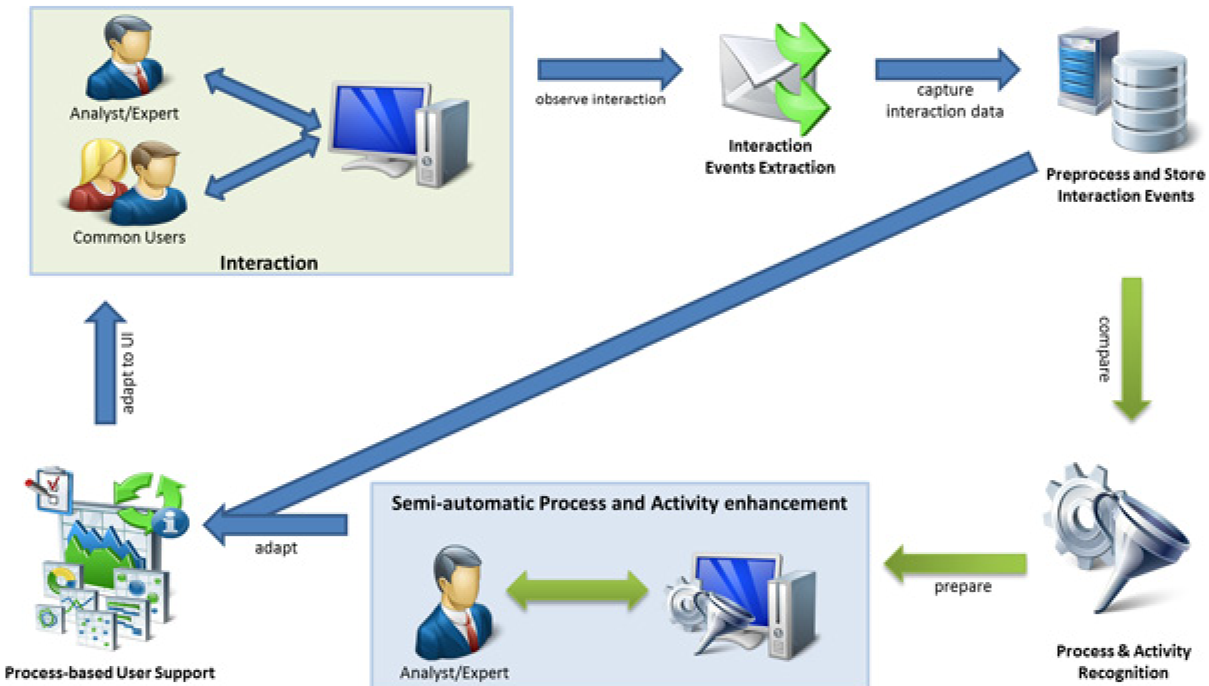
General Model

The six-step model we defined contains two different phases. The first phase addresses the process generation and the second phase addresses the later following adaption of the user-interface. In general we have to distinguish between these two main phases, consisting of the data acquisition for the generation of the process model at the first phase, and the user-interface adaption based on the process model in the second phase.

The general process adaption model describes an iterative process that consists of six phases:

1. Interaction phase: During the interaction phase the users work with the system. At the very beginning no process adaption is being considered. Therefore, the major target user group should consist of experts/experienced users, who know the system and the workflow to solve their tasks. Later – after the process model has been generated – also non-experts can work with the system, because of the active supporting features.
2. Interaction events extraction phase: During the regular interaction, beginning by launching the application, the visualization system generates several events. These events can be subdivided into different types: (i) system events that inform about some technical system states, e.g. a configuration was read, (ii) status events, which inform about changes due to the user's interaction, e.g. a click on a graphical node or a menu entry, (iii) abstracted (high-level) events with a basic semantic meaning, e.g. selection of an entity as logical consequence of a mouse click on a graphical node. All of these events will be generated during the runtime of the system, and they are extracted for further probabilistic analysis.
3. Preprocessing and storing interaction events phase: In the phase of preprocessing and storing of the extracted events the retrace-ability is in focus, so that performed interactions with regards to the correct order can be followed. This means that the events are stored with respect to the session, the correct order, and the involved user. This is necessary to mine a valuable process. Moreover some

Figure 2. Generalized interaction and activity processing model to support the user during his task. The blue arrows indicate the regular interaction cycle, whereas the green arrows indicating the process mining and editing phase of the system. The regular interaction cycle is used for every user interaction, whereas the process mining and editing phase runs only in longer time intervals.



continuing preprocessing can be done to reduce the mining complexity. But such pre-calculations depend on the used algorithms.

4. Process and activity recognition: The process and activity recognition phase is the main processing step. In this phase, the system automatically mines and computes the process based on the collected events. The resulting process model works fully autonomously, but it describes the process and contained steps only in a very technical manner, e.g. input text or press button. The reason for this circumstance is the limited semantic information from the events. Here, only basic information entities are available, but not their precise meaning – however in a limited way a meaning can be determined, e.g. selection of an entity. Because of this fact, the mined process definition needs to be refined from an expert to specify and label the contained steps more specifically.
5. Semi-automatic process and activity enhancement: After the automatic process and activity recognition, an expert has to refine the generated process. The minimal effort an expert has to do is to rename the activities with human-readable labels. In the worst case, the expert has to restructure or enhance the process cycle. The required enhancement effort of the expert depends on the used process mining algorithm. Overall, the expert has all possibilities to make changes on the process. Furthermore, he has to take care about defining a valid and functional process, because based on the applied process the user will be guided through the tasks later. The better the process is defined, the better the usability und use

for the users. But also the adaption abilities provide a better support and joy of use, if the process is as well specified as possible.

6. Process-based user support: In the last phase, the user is supported in his interaction. Therefore the generated process including the adaption functionalities are used to support the user in solving the tasks. The support is based on the current phase in the process. The support of the user can be enhanced, if the role of the user or a kind of personalization is also considered.

The presented overall process introduces all relevant steps. For the common use, it is not always necessary to generate a new process. The semi-automatic process mining steps 4 and 5 are only performed between longer time intervals. We can split the described approach into two major parts: (1) the process mining phase, including the initial learning phase, and (2) the practical use phase with the adaption based on the mined process.

The Process Mining Phase

The most analytical part is the process mining phase, in which the event data is transformed to mine/model a process. In general, it does not matter what kind of process model is targeted. So, among others, it can be, for example, a Petri net or a BPMN model (Van der Aalst, 2011) – we focus on the latter, because it allows a better representation for visualization purposes. In general, we can describe the mining in very simple representation.

The simple event Log L, with a set of traces over a set of activities A for which $L \in B(A^*)$ holds, and a function (or its implementation an algorithm) Y that maps the log L onto a BPMN model:

$$\mathrm{Y}\left(\mathrm{L}\right) = (\mathrm{N}, \mathrm{M})$$

Furthermore, we see the recorded events not only as local independent states. Instead, we understand them as frequently and sequentially generated entities during the interaction process. To respect this fact sufficiently, we focus on heuristic mining methods (Weijters & Ribeiro,2010; Weijters & van der Aalst, 2003; Van der Aalst, 2011, pp.193). In contrast to the simple approaches, such as the $\acute{a}$-algorithm (Van der Aalst, Weijters & Maruster, 2004;Van der Aalst, 2011, pp.129) is that rarely used paths should not be incorporated into the model. This representation bias provided by the causal nets and the usage of frequencies makes the approach more robust than most other approaches (Van der Aalst, 2011, pp.163).

All of the existing approaches aim at generating a process model that is based only on the event logs. Through the consideration of heuristic mining methods, also the sequence and the frequency of their incidence will be considered and supports the generation of an improved process model. In the current version we are using a common heuristic mining method, with just some minor changes which were necessary to integrate it into our system and based on the features of the events. In future work, we expect to make some sophisticated changes to improve the mining process and the generated process model.

Assistance and Recommendations

In the second phase, we use the process model to assist the user in his work on the visualization system. Therefore, the process model acts as an orientation on what the user can do in each phase of the process. Because of the fact that the process model is based on events, each activity can be recognized based on the sequence of events. This automatically allows recognizing the current task and activity of a user.

The most basic support feature is to support the user by provisioning a view in which phase of the process he is currently working. Beyond that, approaches can also be aligned to a single task. For instance, if a user interacts in an analysis task, statistical visualization can be recommended. In fact, the user gets those visualizations that are most appropriate to solve tasks, but he can always choose other visualizations, which are more appropriate or a personal preference. The general architecture is sketched in *Figure 3*. The core of this process adaption model is the process control, which observes all system events and compares it with the process model. In this manner, the system knows about the current task and activity and can assist the user through enabling of supporting features. Supporting features are in the most basic version visualizations, but we also understand features such as guidance or hint functions, as well as other recommender or adaption functionalities.

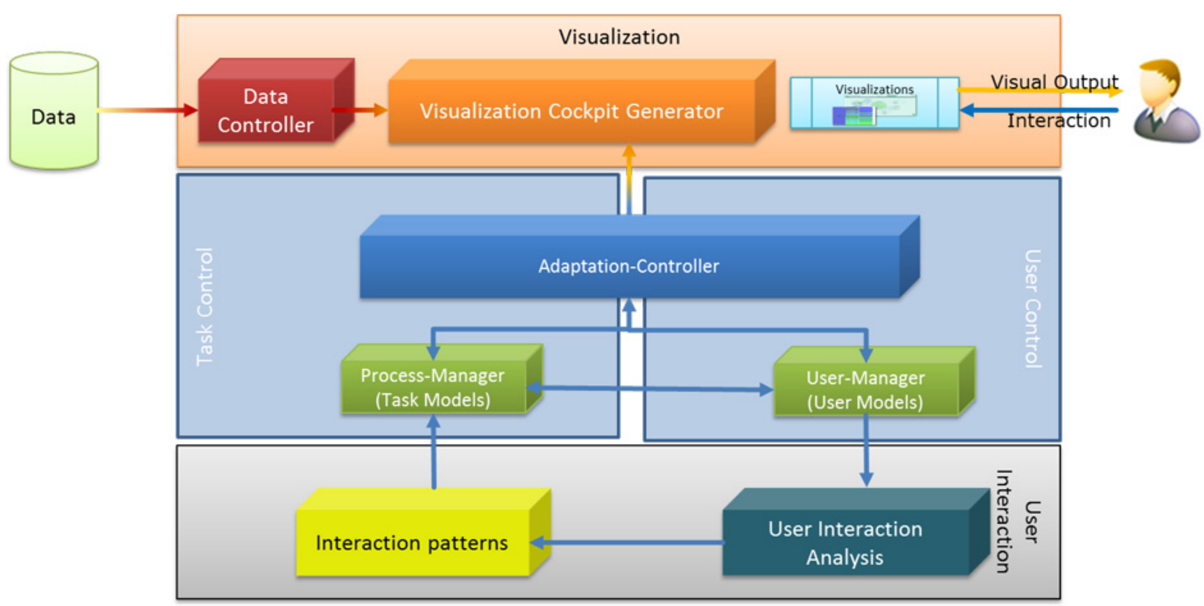
Aside from the process model, also a user model can be used to consider the user's behaviors too. As a result, visualizations can be parameterized to improve the presentation of the data in perspective to the user needs.

A special characteristic of our approach is that, on the one hand it uses all kinds of internal events to recognize the current task of a process, but even more important is the weighted consideration of interaction events. The reason for this weighting of the user interaction is that the user being the main actor in the visualization system can choose how and what task he wants to accomplish. Only for improved, active task recognition the internal events (such as information of the current use a specific data-model or that certain visualization was enabled) will be used next to the interaction events.

IMPLEMENTATION

The implementation follows the described concept. As technical foundation we use the SemaVis

Figure 3. General concept for a dynamic process-supporting system based on user's behavior. Therefore, it allows integrating process supporting features to adapt the user-interface based on the current task and activity. Furthermore it enables the inclusion of a user analysis module to consider the user's behavior in the context of the current task and activity.



framework[1], which is an information visualization system that shows information from heterogeneous data-sources in a graphical manner. This client technology is compatible to the Flash player, builds the basement of our implementation of dynamic user-supporting adaptation concept.

From the technical perspective, the presented concept bases on two parts. The first part is the process analysis web-service, which is majorly responsible for the process-mining and therefore the calculation of the process model. The visualization client sends all events to the web-service, which stores the events in a database. Afterwards a (Process-) Analyst/Expert runs the semi-automatic process mining and enhances the process model. The result is a very detailed process model, which is used by the second part, the visualization client (in our environment we use SemaVis). The process model supports the user during the interaction with the visualization client through enabling of supporting features. In a simple version the user gets shown process visualization for his current task, so that he has an orientation about the outstanding steps to finalize the task. In an advanced mode the visualization can guide the user, e.g. through automatic enabling of visualizations or other tools.

In addition to the process support, it can also be combined with user-adaption techniques, such as the user-based adaption by Nazemi, Stab & Kuijper (2011). This allows for adapting the user-interface also on the user's behavior. The described approach of Nazemi et al. aims to change the visual primitives to highlight the relevant entities of the user based on the user's interaction history. Therefore, the user gets an optimized user-interface based on the current task on the one hand, and based on the personal behaviors on the other hand.

The first prototype is shown in Figure 5. In the top (1) the user can perform a search query and choose the data-source. On the right side (2) the user can choose a visualization, which will be placed and can individually arrange on the visualization desk (3). The process support module is running hidden in background. Currently only the status is shown by workflow visualization (4), and some visualization will automatically be opened in dependence of the current process step. This prototype will be enhanced with a couple of further policy making features, which also requires extending the process model.

Figure 4. An overview of the entire Process Analysis Architecture. In general, multiple instantiations of the visualization client are used at the same time. All events on the system are sent to the process analysis web-service, which generates a process model semi-automatically. Based on the process model the user gets actively supported during his further system use.

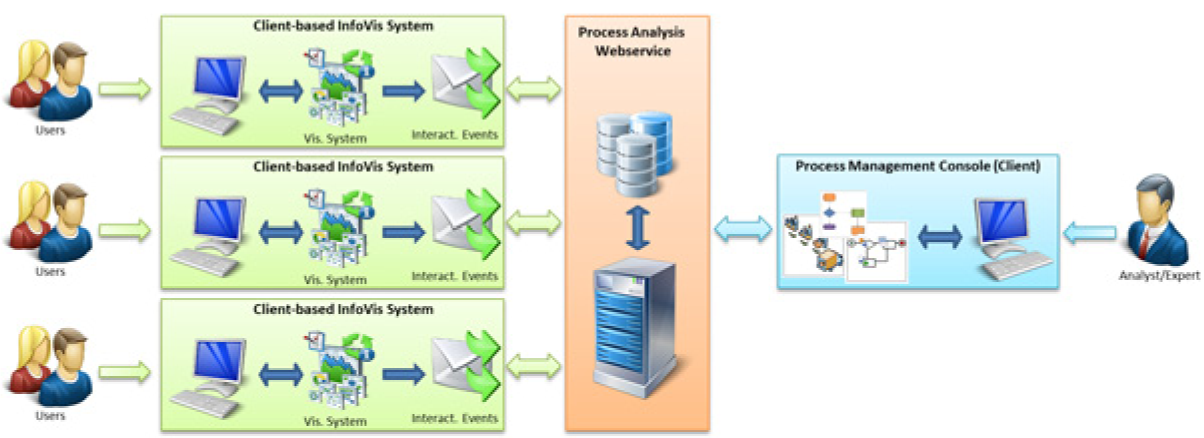
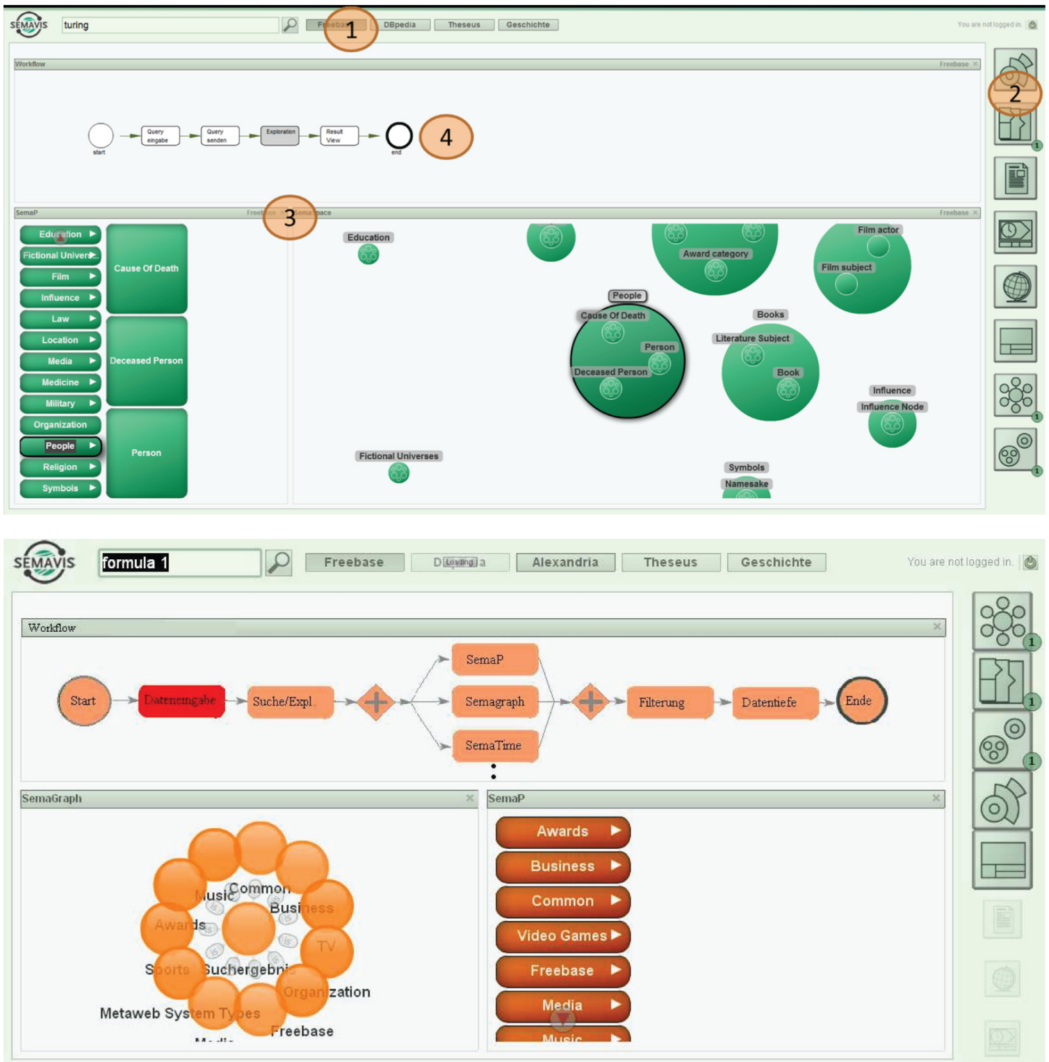


Figure 5. Two screenshots of our first prototype of the process supporting visualization system. The process visualization shows the current active step within the process of semantic data exploration. In the screenshot on the bottom a more extensive, but with less details defined, process is used.



CONCLUSION

In this chapter we introduced into different supporting techniques, which we grouped into seven categories. We introduced into their dynamics, their strictness and in consequence how the interaction of the user gets more or less limited. These features build the basement of our concept, where we introduced a dynamic supporting approach to assist users through complex processes. The goal is to guide or support users through complex processes by enabling or disabling various features depending on the current process phase and the user needs. This ensures a more effective guidance of users' through the task solving routine in dependence of their experiences and knowledge. So on the one hand, the experts are not limited in the way of how they use and interact with the system, but on the other hand, novices guided rather strictly through the task solving routing, so that they will not fail in solving their tasks.

In contrast to the introduced features, our approach considers them as technological feature to assist the user in his work. But next to the mentioned supporting techniques, we also use common techniques, such as specifically designed visualizations which consider the requirements on the user (e.g. multi-dimensional visualization for experts and common visualizations for novices), or editors to make changes on data. The main benefit of the resulting system is an intelligent system that enables stakeholders of the policy modeling process to do their tasks more efficiently and effectively.

REFERENCES

Appert, C., & Zhai, S. (2009). 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.

Burkhardt, D., Breyer, M., Glaser, C., Stab, C., Nazemi, K., & Kuijper, A. (2011). 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 (LNCS) (vol. 6765, pp. 20-29). Berlin: Springer.

Burkhardt, D., & Nazemi, K. (2012). Dynamic process support based on users' behavior. In *Proceedings of 15th International Conference on Interactive Collaborative Learning (ICL2012)*. IEEE.

Burkhardt, D., Ruppert, T., & Nazemi, K. (2012). Towards process-oriented Information Visualization for supporting users. In *Proceedings of 15th International Conference on Interactive Collaborative Learning (ICL 2012)*. IEEE.

Burkhardt, D., Stab, C., Steiger, M., Breyer, M., & Nazemi, K. (2012). Interactive Exploration System: A User-Centered Interaction Approach in Semantics Visualizations. In Proceedings of Cyberworlds. Darmstadt, Germany: Academic Press.

Charrada, E. B., & Glinz, M. (2010). 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.

Gaul, W., Geyer-Schulz, A., Hahsler, M., & Schmidt-Thieme, L. (2002). eMarketing mittels Recommendersystemen. In Marketing–Zeitschrift für Forschung und Praxis (ZFP), (pp. 47–55). Academic Press.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Magazine of the ACM, 35(12), 61-70.

Günther, C., Schönig, S., & Jablonski, S. (2012). Dynamic guidance enhancement in workflow management systems. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*. ACM.

Hjert-Bernardi, K., Melero, J., & Hern'ndez-Leo, D. (2012). 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). IEEE.

Johnson, J. (2010). *Designing with the Mind in Mind: Simple Guide to understanding User Interface Design Rules*. Burlington, MA: Morgan Kaufmann.

Lin, E., Greenberg, S., Trotter, E., Ma, D., & Aycock, J. (2011). Does domain highlighting help people identify phishing sites?. In *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*. ACM.

Nazemi, K., Stab, C., & Kuijper, A. (2011). A Reference Model for Adaptive Visualization Systems. In Human-Computer Interaction: Part I: Design and Development Approaches. Berlin: Springer.

Neumann, A. W. (2008). *Recommender Systems for Scientic and Technical Information Providers*. (Dissertation). University at Karlsruhe (TH).

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.

Ostkamp, M., Bauer, G., & Kray, C. (2012). Visual highlighting on public displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis '12)*. ACM.

Pohl, K., Weidenhaupt, K., Dömges, R., Haumer, P., Jarke, M., & Klamma, R. (1999). PRIME - toward process-integrated modeling environments. *ACM Transactions on Software Engineering and Methodology*, *8*(4), 343–410. doi:10.1145/322993.322995

Resnick, P., & Varian, H. R. (1997). Recommender systems. Magazine of Communications of the ACM, 40(3), 56-58.

Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008). Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction (TEI '08)*. ACM.

Silingas, D., Pavalkis, S., & Morkevicius, A. (2009). MD wizard - A model-driven framework for wizard-based modeling guidance in UML tools. *Proceedings of International Multiconference on Computer Science and Information Technology, 9*, 609–615.

Sliski, T. J., Billmers, M. P., Clarke, L. A., & Osterweil, L. J. (2001). An architecture for flexible, evolvable process-driven user-guidance environments. SIGSOFT Softw. Eng. Notes, 26(5), 33-43.

Torre, I. (2009). Adaptive systems in the era of the semantic and social web, a survey. Journal of User Modeling and User-Adapted Interaction, 19(5), 433-486.

Van der Aalst, W. M. P. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Heidelberg, Germany: Springer. doi:10.1007/978-3-642-19345-3

Van der Aalst, W. M. P., Weijters, A. J. M. M., & Maruster, L. (2004). Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, *16*(9), 1128–1142. doi:10.1109/TKDE.2004.47

Weijters, A. J. M. M., & Ribeiro, J. T. S. (2010). *Flexible Heristic Miner (FHM)*. Eindhoven University of Technology.

Weijters, A. J. M. M., & van der Aalst, W. M. P. (2003). Rediscovering Workflow Models from Event-Based Data Using Little Thumb. *Integrated Computer-Aided Engineering*, *10*(2), 163–190.

Wigdor, D., & Wixon, D. (2011). *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann. doi:10.1016/B978-0-12-382231-4.00004-6

KEY TERMS AND DEFINITIONS

Adaptation: Adaptation in human-computer interfaces is the automatic and system-driven changes on content, structure, and presentation of system-behavior that involve some form of learning, inference, or decision making based on one or many influencing factors to support users.

Adaptive Visualizations: Adaptive visualizations are interactive systems that adapt autonomously the visual variables, visual structure, visualization method, or the composition of them by involving some form of learning, inference, or decision making based on one or many influencing factors like users' behavior or data characteristics to amplify cognition and enable a more efficient information acquisition.

Information and Communication Technologies (ICT): Under Information and Communication Technology technologies for information provision, sharing, using and visualizations are summarized. A major benefit lays in the exchange of data for the use with other technologies and therefore the use in a number of different use cases.

Policy Modeling: The term policy modeling deals with the making of (political) policies, which can result in the creation of new laws. Policy modeling covers all necessary steps beginning at the identification of a problem, analysis, decision making, implementation, end evaluation of a policy.

Policy: Under the policy a theoretical or practical instrument can be understood that aims to solve a specific problem. In the political domain, a policy can represent a new law.

Process: A process is logical aggregation of activities. A process is defined by an initial state and an (to achieve) end state. Through the processing of the activities, the transformation from the initial state to the end state will be realized.

SemaVis: SemaVis[i] is an adaptive semantics visualization technology developed by Fraunhofer Institute for Computer Graphics Research.

ENDNOTES

1 More information about SemaVis on: http://www.semavis.com.