Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2013 Completed Research

ECIS 2013 Proceedings

7-1-2013

Visualising Process Model Hierarchies

Kathrin Figl
WU - Vienna University of Economics and Business, Vienna, Austria, Austria, kathrin.figl@gmail.com

Agnes Koschmider
Karlsruhe Institute of Technology, Karlsruhe, Germany, agnes.koschmider@kit.edu

Simone Kriglstein
SBA Research, Vienna, Vienna, Austria, skriglstein@sba-research.org

VISUALISING PROCESS MODEL HIERARCHIES

Figl, Kathrin, WU - Vienna University of Economics and Business, UZAII, Augasse 2-6, 1090 Vienna, Austria, kathrin.figl@wu.ac.at

Koschmider, Agnes, Karlsruhe Institute of Technology, Institute AIFB, Building 05.20, 76128 Karlsruhe, agnes.koschmider@kit.edu

Kriglstein, Simone[1], SBA Research, Favoritenstraße 16, 1040 Vienna, Austria, skriglstein@sba-research.at

Abstract

In this paper we seek to expand our understanding of how to visualize process model hierarchies. While the information systems community has devoted considerable attention to decomposition as a principle to manage complexity of conceptual models, surprisingly little research has explicitly addressed two relevant factors for understandability of model decomposition in a tool– how the model hierarchy is visually represented (visualisation) and how users can interact with the hierarchical structure (interface). Using an expert evaluation approach, the paper, therefore, aims to determine whether some visualisation strategies provide a better fit to model hierarchies than others. Based on an evaluation of an example of a process model hierarchy visualisation, the results of this study indicate that experts would prefer to use a node-link visualisation over nested graphs and treemaps and to navigate in the hierarchy with the help of an overview+detail instead of a focus+context strategy. These insights can be used to develop user-centred modelling tool support. Finally, a broader goal of this study is to stimulate discussion about the relevance of visualisation techniques for understanding model decomposition.

Keywords: Process Models, Decomposition, Visualisation.

[1] The research was funded by COMET K1, FFG - Austrian Research Promotion Agency.

1 Introduction

Process models are relevant for understanding and communicating a domain in order to support development of information systems and to define and document requirements. As business process models are "simplifications in order to bring clarity and understanding to some aspect of a problem where there is complexity, uncertainty, change or assumptions" (Lindsay et al., 2003, p. 1018), their visual representation should not place extra burden for understanding them, but rather support comprehension.

In practice, business process models can get very complex and understanding of large process model collections is a challenge for the information systems discipline. Representation of large processes in a single, monolithic model might easily lead to a "map shock" in model viewers (Moody, 2006a). This "feeling of being overwhelmed by the scale and complexity of a map display" might in turn reduce motivation to work with a model (Blankenship and Dansereau, 2000, p. 295). Thus, the large size of models makes it necessary to manage complexity, which is usually realized by the concept of modularization, also called decomposition (Moody, 2006b). Decomposition is not a new problem, Courtois (1985, p. 590) has already mentioned in 1985 that "models of large and complex systems can often be reduced to smaller submodels, for easier analysis, by a process known as decomposition." Decomposition can be defined as "the breakdown of a complex system into smaller, relatively independent units" (Paulson and Wand, 1992, p. 174).

Modularisation yields another problem: as a variety of models and submodels are produced, users have to assimilate different pieces of information from multiple models simultaneously and cognitively integrate them (Moody, 2006b, Kim et al., 2000). This in turn, as Zugal et al. (2012) point out, may lead to further cognitive load for users because of a split-attention effect (Sweller and Chandler, 1994), especially if user are not well supported in navigating between different hierarchy levels by the information visualisation. Thus, it is important to use different visualisation strategies that support the user in integrating the information parts.

Zugal et al. (2012) conclude from a literature study on prior empirical investigation of the effect of modularization on process model comprehensibility that it "remains unclear under which circumstances positive or negative influences can be expected". However, research on other types of conceptual modelling as UML statechart diagrams (Cruz-Lemus et al., 2009) demonstrated general benefits of modularization. One of the limitations of existing studies is that they fail to address the moderating effects of the design of the visualisation when users interact with the different models and submodels. A more systematic study on the benefit of process model modularization should take different visualisations of process model hierarchy into account and might shed light on inconsistency of previous findings. As in practice users usually use modelling tools, especially when creating and edited process models, it is relevant how models and submodels are visually presented and how the user is supported in orientating and navigating through the model structures. Interactive visualisations techniques may help to represent large process models on limited screen space, prevent the user of losing overview in submodels and help him/her to mentally integrate information from different submodels (North, 2005). In our argumentation we follow Woods (1984, pp. 229-230) who argues on performance losses when users work with information on various screens that "the 'getting lost' and 'keyhole' phenomena are not inevitable consequences of using computer-based displays; neither do they represent human limitations (for example, short-term memory) … difficulties are the result of a failure to consider man and computer together as a cognitive system". Human cognitive benefits of modularization approaches have to be combined with computer-based visualisation techniques to reach their full potential to help users to understand process models.

Our motivation is to complement the stream of work on process model decomposition by examining the role of visualisation strategies for hierarchies. The remainder of this paper proceeds as follows. The theoretical contribution of this paper is twofold – first, we review literature on hierarchical design of process models, second, we turn to visualisation of hierarchies. We discuss visualisation strategies as node-link diagrams, treemaps and nested graphs as well as interface strategies as overview+detail and focus+context. The paper then provides practical insights by exploring how these visualisation techniques can be used for visualising hierarchical process model structures. The next section discusses the design of the expert evaluation and the measurements used, before we present our data analysis and an examination of the results. We conclude this paper by providing a short discussion and conclusion.

2 Hierarchical Design Structure

2.1 Concepts, Terms and Techniques

Several concepts have been adopted from Software Engineering in order to reduce complexity of business process models (in the sense of scaling the size of the process model) and to increase understandability. In this paper we refer only to the level of process activities and disregard any other business model (e.g., organizational/role or object model). In Software Engineering, *modularity* and *abstraction* are seen as key principles for “successful” (component-oriented) software development (Aßmann, 2003). In the design of business processes, the concept of modularity can be applied through the use of subprocesses. Quality criteria to consciously decompose a system into modules have been for instance discussed by Wand and Weber (1995). A general goal in decomposition is to achieve high cohesion within submodels and low coupling between submodels (Paulson and Wand, 1992).

However, a modular designed business process does not necessarily implement a hierarchical design structure (see Parnas, 1972). A business process can be splitted in two disjoint business processes describing procedures on the same granularity level. In this paper we presume a general occurrence of modularity, thus advocating a hierarchical design, which is also in line with the fact that in business practice “hierarchical decomposition plays a central role for organizing processes in an understandable way and for refining coarse-granular towards a fine-granular representation” (Malinova et al., 2013). Due to a clear relationship between the size of a process model (level) and the use of modularity it is recommended to decompose a process model if it has more than 50 elements to reduce the risk of errors and improve its understandability (Mendling et al., 2010).

In general, when decomposing (respectively composing) a business process, two modelling styles can distinguished (Koschmider and Blanchard, 2007). In top-down modelling (decomposability), the top level process formulates an overview of process elements, without providing detailed descriptions of process elements. Consequently, the top view of the process is structured in a more fine-grained way by refining process activities to subprocesses. By using the bottom-up approach (composability), in contrast to the top-down approach, modelers start modelling more specific process elements, which are subsequently linked together to coarse-grained processes. The linking or coarsening is done until a complete abstract view of the process model is achieved. A plethora of approaches is available for decomposition of process models. For instance, a method for automating the process of systems decomposition is presented in (Paulson and Wand, 1992). Courtois (1985) applies formal and statistical method to accurately control the level of approximation when using decomposition techniques.

Top-down and bottom-up modelling assume an abstract view of a process model. Abstraction is understood as the separation of important details from the unimportant ones and allows focusing on key activities. Two implementation principles of abstraction are generalization and specialization. Generalizations and specializations can have two types of constraints: (i) the disjoint/overlap relationship constraint (IS-A), and, (ii) participation constraints – total or partial (IS-PART-OF) (Elmasri and Navathe, 2007). Business process model specialization can be used to describe a process model as a type of another process model (Malone et al., 2003). Business process model abstraction can be variously implemented. Polyvyanyy et al. (2008) suggest the abstraction of business processes by using specific criteria (relative effort of an activity) for aggregating activities. An abstraction approach based on the model behaviour is presented by Weidlich et al. (2010) and Polyvyanyy et al. (2009) uses a tree-based representation of process models that are continually abstracted. Different use cases for business process model abstraction are consolidated in (Smirnov et al., 2012).

Consequently, hierarchically designing business processes demands to appropriately use decomposition and abstraction/specialization (Malone et al., 2003). Figure 1 summarizes the principles of a hierarchical design of business process models.

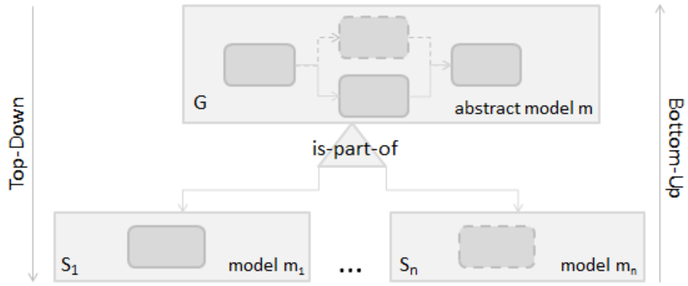


Figure 1. Hierarchical design of business process models.

The question on how decomposition is used in process models in practice closely relates to process architecture. Despite the relevance of process architecture to organize all process models used in an organization, there is little empirical evidence on how companies decompose process models. An exception is a qualitative study conducted by Malinova et al. (2013). Based on their data set they concluded that companies typically use at least three levels of granularity and that the first level consists of unconnected main processes, often called "process land card". In addition, their study demonstrated that depending on the company type, different archetypes of decompositional process architectures can be found. For instance, in "hierarchical process architectures", sub-processes relate to exactly one process model on a higher level. In a specific form of hierarchical process models the "pipeline process architectures", which can be found for instance in a manufacturing context, the first level processes form a chain of execution. In contrast to traditional hierarchical architectures, in "service-oriented process architectures" reuse of fine-granular process models is a central goal and as a consequence, decomposition is not strict, but fine-granular process models can belong to more than one higher-level process model.

2.2 Support for Hierarchical Process Design in Practice

To evaluate how the hierarchical design is supported in practice, we compared 31 tools, 27 of them are maintained in the "Magic Quadrant for Business Process Management Suites" by Gartner (Sinur and Hill, 2010). We were interested to investigate how (and if) a hierarchical design (decomposition, abstraction, generalization, specialization) of process models is supported. The analysis was part of a larger project on evaluating process model reuse in BPM tools (for more information on the tool analysis we refer to Koschmider et al. (2013)).

All tools allow defining subprocesses. A newly created or archived process model can be selected and defined as a subprocess. The modelling can be started from top-down (e.g. with a process land card) or processes can be created bottom-up by assigning process models to an activity of a parent process (decomposability and composability are supported). However, a subprocess model mostly cannot be assigned to an additional process model activity. Thus, reuse of subprocesses is difficult.

Almost all tools visualise the hierarchy of process models in a tree list (or explorer navigation, see Figure 2). A double click on a process model name (e.g., "Process order end to end") opens the corresponding visual view of the process model (usually in the diagram workspace on the right hand side). Only one BPM tool (process modeler of BizAgi) offers in addition to the tree list a visual preview of all archived process models in a miniature view. This feature allows the user to get an overview of all process models without sequentially opening each process model. Another overview solution is offered by the BPM tool of Cordys (BOP4), which provides a three-dimensional view of all opened diagrams (similar to the 3d view of tabs in Windows).

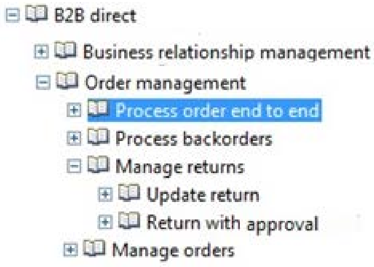


Figure 2. Tree list to visualize a process model hierarchy.

To understand relationships between a process model, its subprocesses, variants and related business models, the user typically has to manually navigate through the diagrams. None of these tools graphically visualise connections between process models and its predecessor versions. Some rudimental support for abstraction (abstraction slider) is offered by the BPM tool of Signavio.

3 Visualisation and Interface Strategies for Process Model Hierarchies

One main result of the tool evaluation presented in Section 2.2 was that for visualising process model hierarchies tree-based menu lists predominate. This section discusses further approaches for hierarchical visualisation as well as interface strategies from a theoretical perspective. First, we present approaches to display hierarchical relationships between objects (viz. between process and subprocesses). Second, we turn to interface strategies to navigate between a process model and its context in the hierarchy.

3.1 Visualisation

In general, the two representations *link* and *containment* are distinguished (North, 2005). A very popular type of the link visualisation is the *node-link* representation (see e.g. Ghoniem et al., 2004). Often used representative types for containment representation are *treemap* and *nested graph*. Figure 2 exemplarily shows the three visualisation types node-link, treemap and nested graph.

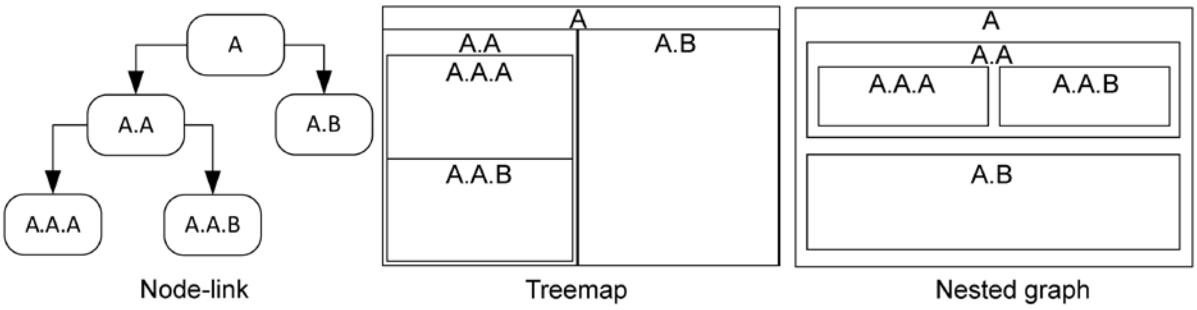


Figure 2. Strategies for the visualisation of hierarchical relationships.

The node-link representation is one of the most frequently used approaches to visualise relationships between business processes in general (see, e.g., Moody, 2009, Diguglielmo et al., 2002, Effinger et al., 2009). Therefore, this kind of representation has the advantage of being very familiar to users if it is also used for visualising the process model hierarchy. The objects are visualised as nodes and edges represent the connections between the objects. The node-link visualisation directly shows the connections between objects and provides a good overview about the structure. However, to effectively visualise node-link approaches, space is needed in order to be able to consider aesthetic principles, like minimization of edge crossings and minimization of node overlapping (Card et al., 1999, Ware, 2004, North, 2005). For large graphs, the representations can still be cluttered, because of the limited screen space.

Treemap is a space filling and containment approach and can be used as an alternative to link approaches. In general, space filling approaches use the whole screen space by recursively subdividing the space based on the node and their children (Shneiderman, 1992). The hierarchical relationships between objects are visualised in such a way that the objects visually contain their subobjects. Treemap represents objects as rectangles and the hierarchical relationships result from alternate horizontal and vertical subdivision into smaller rectangles, which represent the subobjects. The advantages of treemaps are that the hierarchical structure is completely visible on the available screen space and allows jumping between the objects without losing the orientation. However, a treemap representation is usually unfamiliar to users and they have to learn to interpret the visualisation correctly (Babaria, 2001). Furthermore, if an object is connected to multiple objects (e.g., a subprocess has more than one parent process), this object has to be placed within all its parents. Therefore it can happen that it is not always clear to the user why the same object is visualised two or more times. Moreover, for large and deep trees, the reduced space between the internal nodes makes it more difficult for the user to reconstruct the information about the hierarchical structure (Ham and Wijk, 2003).

Nested graphs are similar to treemaps in so far that both approaches nest the sub-process in their process. But, in contrast to treemaps the hierarchical structure is clearer visible because of the additional space between the internal nodes. On the other side, this additional space leads to the drawback that nested graphs do not use the available screen space as effective as a treemap. Similar to treemap, the nested structure allows users to gain insight on relationships between process models. Nest graph and Treemap share the disadvantage that subprocesses have to be placed within all its parent processes for the case that they are connected to more than one parent process.

3.2 Interface Strategies

Interface strategies are necessary to support the coordination between a particular process model and its context and can be realised via a *focus+context* or an *overview+detail* approach.

The focus+context approach concentrates on integrating the process model into the hierarchical visualisation of process models within one single window. Focus+context strategies are a “natural match for tree navigation, enabling users to drill-down within an individual branch of focus in the tree while maintaining context of the path” (North, 2005). In such a scenario, the detail information is visible with help of techniques like distortion or elision. Distortion techniques spatially expand the focus region directly within its context (North, 2005). The disadvantage of such distortion techniques is that in case of extreme distortion the valuable contextual information is hidden. A variety of authors have presented proposals how the popular fisheye distortion technique, which reminds of a magnifying glass can be applied to zoom conceptual models (see, e.g., Summers et al., 2003, Storey and Müller, 1996, Frisch et al., 2008). Elision techniques hide parts of a structure and show them only on demand (Ware, 2004). Such techniques can be applied to hide subprocesses in their process model in such a way that only their abstract label is shown until the subprocesses move into the main point of interest. Polyvyanyy et al. (2008) propose to use a semantic zooming technique called “process model abstraction sliders”, which could be characterized as elision technique.

The overview+detail strategy uses multiple windows that separately display a particular process model and its relationships. A disadvantage of the overview+detail strategy is that focus and context information is not directly linked and has to be mentally integrated (Ware, 2004). However, it simplifies the design and can provide undistorted views on focus and context information.

4 Expert Evaluation of Visualisation Approaches

While there already exist a variety of reviews on usability and user preferences of different visualisation and interface strategies (e.g. Cockburn et al., 2009), results vary across different domains and task settings and therefore, their fit for process model hierarchies has to be evaluated empirically. Thus, it remains to be evaluated what kind of hierarchical visualisation suits well to process modellers. To address this research question, we developed a web-based questionnaire to collect expert evaluations on the visualisation and interface strategies. We briefly describe important elements of the online questionnaire in the following. First, the online system explained in a “participant information” section the objective of the study and asked participants for their consent in completing the questionnaire voluntarily and anonymously. The next section asked participants how many (process) models they had read or created up to date and whether they had received any training in (process) modelling. In addition, it collected data on basic demographics (age, gender, education etc.). The third section presented three different hierarchy visualisations to participants and confronted them with four comprehension tasks for each of these visualisations to assure that participants would get familiar with them. After showing each visualisation to participants, we asked them to rate perceived ease of use and perceived usefulness and to indicate their preferences for using one of the visualisations over another. The fourth section introduced participants to the two interface strategies overview+detail and focus+context and showed them visual examples. The system automatically proceeded after a time period of 10 seconds (to view each combination of the interface strategies with the three process visualisation) elapsed. In the end, the section included preference items for the two interface strategies.

4.1 Materials

To introduce the visualisation and interface strategies to study participants, we used an example of a process model hierarchy of the business to business (B2B) domain. Figures 3, 4 and 5 depict the process model hierarchy visualized as node-link, treemap and nested graph representation. Figure 6 demonstrates the interface strategies overview+detail and focus+context (using a type of fisheye distortion) for the node-link representation; in the online questionnaire participants were also given the same example in combination with the treemap and nested graph representation. We designed the visualisations to ensure they were representative of a typical process model hierarchy. The process model hierarchy has three decomposition levels, whereas each level has at least one subprocess. We also included two examples of reuse of processes – the process “Manage Returns” is used in two parent processes (“Customer Service” and “Order Management”) the processes “Create Contact” and “Renew Contact” are both used in the parent processes “Manage Customer Contracts” and “Manage Base Contacts”. To assure that participants would get familiar with the visualisations to a sufficient degree before evaluating them, we confronted participants with 12 comprehension tasks on the hierarchy (e.g., “‘Update Returns’ is a subprocess of ‘Order Management’” or “‘Revise Contact’ is a subprocess of ‘Manage Business Accounts’”). We randomly assigned four tasks to each of the three visualisations, respectively.

To avoid any order bias, we presented node-link, treemap and nested graph representation in three different orders and also varied the presentation order of interface strategies.

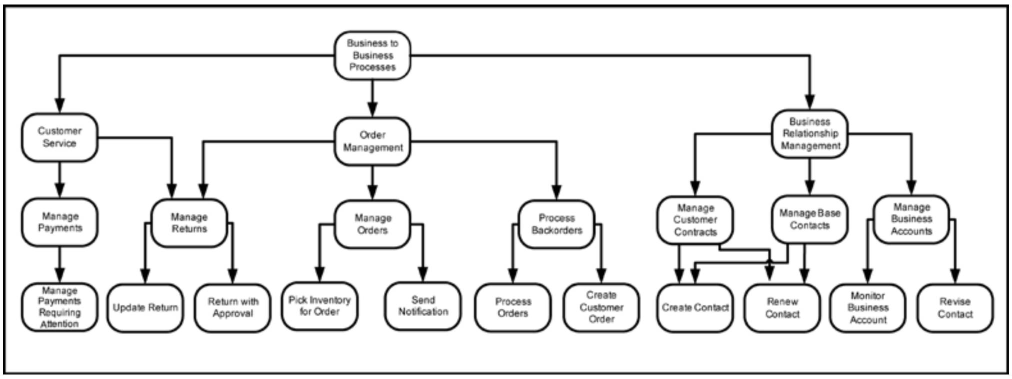


Figure 3. A business process hierarchy in a node-link visualisation.

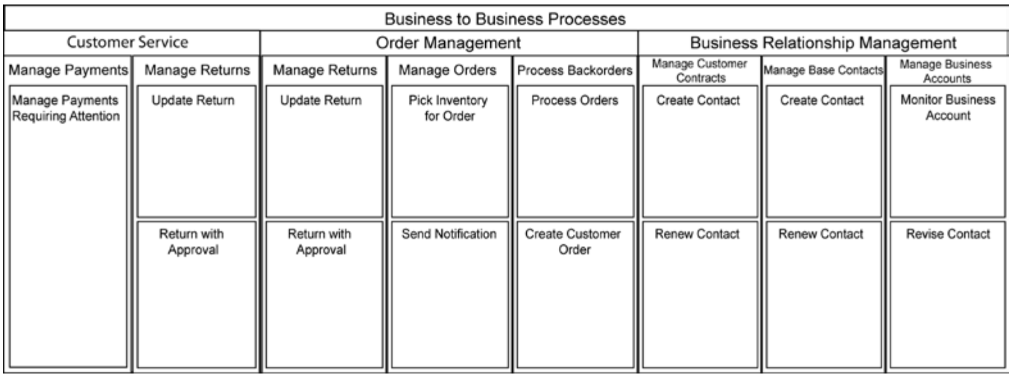


Figure 4. A business process hierarchy in a treemap visualisation.

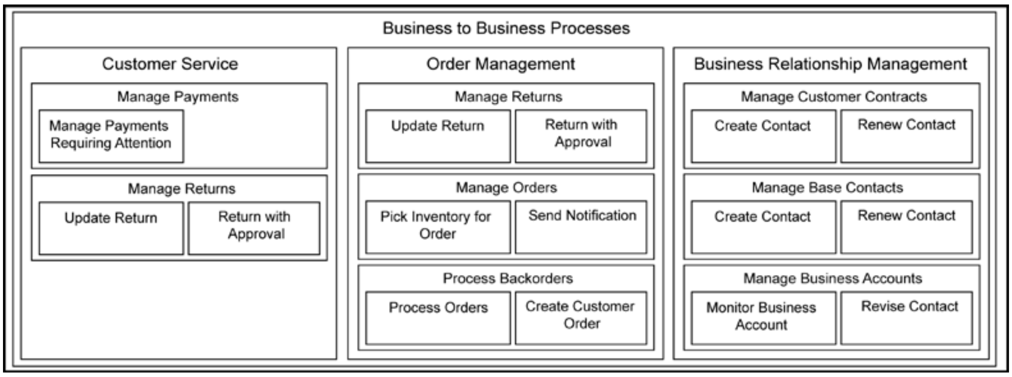
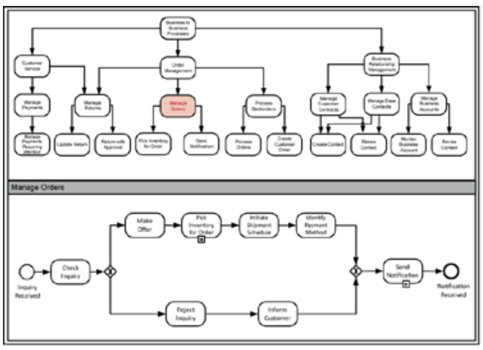
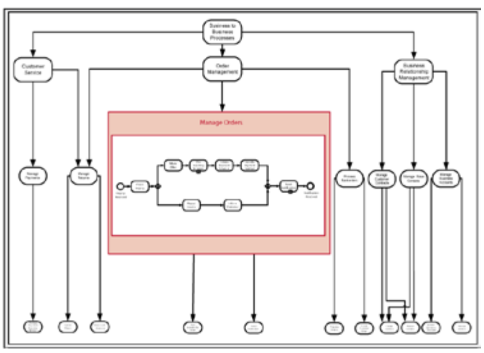


Figure 5. A business process hierarchy in a nested graph visualisation.



Overview+Detail



Focus + Context

Figure 6. Two interface strategies to visualise the context of a process model (its subprocesseses and parent process(es))

4.2 Measurement

We used validated measures for the variables perceived ease of use and perceived usefulness (Maes and Poels, 2007), but replaced the term "conceptual model" with "visualisation" and the term "process" with "business process hierarchy" to fit the items to our evaluation purpose. An example item for the scale perceived ease of use is "Learning how to read the visualisation was easy" and an example for perceived usefulness is "Overall, I think the visualisation improves my performance when understanding the process hierarchy modeled." To estimate internal consistency of these measures for the new wording and our sample, we computed Cronbach's α, which should be greater than or equal to 0.7 to consider items to be uni-dimensional (Nunnally and Bernstein, 1994). Cronbach's α for perceived ease of use varied from 0.82 to 0.91, Cronbach's α for perceived usefulness from 0.92 to 0.98 for rating the three visualisations in our sample, suggesting adequate reliability and internal consistency of measures.

In addition, participants were asked to judge their preference for specific visualisation and interface strategies through pair-wise comparisons. We used the following introduction to these questions: "Imagine your task is to use a process hierarchy visualisation in any way (to model a process hierarchy, to understand it, to explain it to somebody else, or to redesign it) and you can choose one visualisation to assist you." The rating system was implemented using a slider that measured preference for one visualisation/interface over another (on a scale from zero to 100). Specifically, participants were asked to rate

- Preference for *node-link* over *nested graph*,
- Preference for *treemap* over *node-link*,
- Preference for *treemap* over *nested graph*,
- Preference for *nested graph* with *focus+context* over *nested graph* with *overview+detail,*
- Preference for *node-link* with *focus+context* over *node-link* with *overview+detail,*
- Preference for *treemap* with *focus+context* over *treemap* with *overview+detail,*
- Preference for *focus+context* strategy over *overview+detail* strategy in general.

A further single-choice item was used to allow participants to select one out of the three visualisation options they liked best.

4.3 Sample

The population of interest for the data collection were experts in the process modelling field in academia. We recruited such experts sending email advertisements to colleagues working in the process modelling field from different European universities. Overall, 15 completed questionnaires were obtained. As it was an academic expert sample, all participants had already obtained a Masters (8; 57%) or a Doctorate degree (6; 43%) in either information systems (4; 31%) or computer science (9; 69%). Participants were on average 31.5 years old; 43% were female (6), 57% male (8). They already had created 36.64 (median=27.5) and analysed 157.77 (median=30) process models on average.

4.4 Results

To examine whether experts had specific preferences for visualisation and interface strategies we calculated one sample t-tests for each of the sliding scale comparisons for differences from the neutral value 50. Table 1 gives all results of the statistical tests. While experts were indifferent between the visualisations treemap and nested graph, they clearly preferred the node-link visualisation over nested graphs and also tended to prefer it over treemaps. This result is consistent with the fact that in the single-choice option, node-link was selected more often (8; 62%) as best option than nested graphs (2; 15%) and treemaps (3; 23%). However, two repeated measures analyses of variance showed that neither the ratings of perceived ease of use (nested graphs: M=5.01, SD=1.61; treemaps: M=5.27, SD=1.30; node-link: M=5.33, SD=1.54) nor of perceived usefulness (nested graphs: M=5.12, SD=1.66; treemaps: M=5.03, SD=1.38; node-link: M=5.41, SD=1.56) significantly differed between the three visualisations. At least, the mean ratings of perceived ease of use and perceived usefulness showed in the same direction as it was highest for the node-link visualisation. Additionally, we can observe from Table 1 that experts tend to prefer the overview+detail strategy over the focus+context strategy.

Table 1. Statistical test results for expert preferences

Preference	**Mean**	**St. Dev.**	**T (df = 12)**	***p***
node-link over *nested graph*	72.92	33.16	2.49	.028
treemap over *node-link*	29.46	38.59	-1.92	.079
treemap over *nested graph*	44.23	35.75	-0.58	.571
nested graph with *focus+context* over *nested graph* with *overview+detail*	27.09	34.21	-2.22	.051
node-link with *focus+context* over *node-link* with *overview+detail*	28.09	36.94	-1.97	.078
treemap with *focus+context* over *treemap* with *overview+detail*	21.09	28.67	-3.34	.007
focus+context strategy over *overview+detail* strategy in general	27.10	34.35	-2.11	.064

5 Discussion and Concluding Comments

This study set out with the aim of assessing the utility of different visualisation and interface strategies for process model hierarchies. Based on an expert evaluation, our results show that experts prefer a node-link visualisation over nested graphs and treemaps and an overview+detail over a focus+context interface strategy. The negative evaluation of treemaps is consistent with the results of Barlow and Neville (1959); however, in contrast to our results, they had found that compact layouts like nested graphs were at least equally preferred as the node-link layout. In our case the preference of the node-link visualisation could be explained by the fact that process models are also usually represented as node-link diagrams; and thus, the models and their hierarchical structure would be represented consistently according to the same paradigm. The low openness of users for treemaps is surprising, as this visualisation is not more difficult to understand than other visualisations (Kobsa, 2004) and future work is necessary to explain this result. A possible explanation of the lower preference ratings for nested graphs and treemaps could also lay in the fact, that we presented an example hierarchy consisting of 22 process models. User preferences might change for a very large number of process models in the hierarchy; thus, generalisation of our results to larger hierarchies is limited.

Our observation that the overview+detail strategy is preferred over the focus+context strategy confirms the results of Hornbaek and Frøkjær (2003), who have found a strong user preference of overview+detail in a reading task. Our result is also in agreement with the previous finding that hierarchical overview menu designs are superior to focus+context fisheye menu design from a usability perspective (Hornbæk and Hertzum, 2007).

Several possible directions for future research emerge from our study. Further studies could examine visualisations preferences for very large process models hierarchies. In addition, it would be of interest whether process modellers and analysts in practice have the same preferences as the academic expert sample used in this study.

In this paper, we contribute to process modelling research by providing an overview of literature and an empirical analysis of the fit of different visualisation techniques for representing process model hierarchies. Additional research could now extend our work and for instance develop an interactive prototype to enable a usability evaluation of the process model hierarchy visualisations.

References

Aßmann, U. (2003). Invasive Software Composition. Springer.

Babaria, K. (2001). Using treemaps to visualize gene ontologies. Human Computer Interaction Lab and Institute for Systems Research. University of Maryland, MD USA.

Blankenship, J. and Dansereau, D.F. (2000). The Effect of Animated Node-Link Displays on Information Recall. Journal of Experimental Education, 68 (4), 293-308.

Card, S. et al. (1999). Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann.

Cockburn, A. et al. (2009). A review of overview+detail, zooming, and focus+context interfaces. ACM Comput. Surv., 41 (1), 1-31.

Courtois, P.J. (1985). On time and space decomposition of complex structures. Commun. ACM, 28 (6), 590-603.

Cruz-Lemus, J.A. et al. (2009). Assessing the understandability of UML statechart diagrams with composite states--A family of empirical studies. Empirical Softw. Engg., 14 (6), 685-719.

Diguglielmo, G. et al. (2002). Graph Layout for Workflow Applications with ILOG JViews. In Goodrich, M. and Kobourov, S. (eds.) Graph Drawing. Springer Berlin Heidelberg.

Effinger, P. et al. (2009). An Interactive Layout Tool for BPMN. In Proceedings of the Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing, p. 399-406.

Elmasri, R. and Navathe, S.B. (2007). Fundamentals of Database Systems. Addison Wesley.

Frisch, M. et al. (2008). Towards seamless semantic zooming techniques for UML diagrams. In Proceedings of the Proceedings of the 4th ACM symposium on Software visualization, p. 207-208, Ammersee, Germany.

Ghoniem, M. et al. (2004). A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In Proceedings of the Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, p. 17-24.

Greenhouse, S.W. and Geisser, S. (1959). On methods in the analysis of profile data. Psychometrika, 24 (95-112).

Ham, F.v. and Wijk, J.J.v. (2003). Beamtrees: compact visualization of large hierarchies. Information Visualization, 2 (1), 31-39.

Hornbæk, K. and Frøkjær, E. (2003). Reading patterns and usability in visualizations of electronic documents. ACM Trans. Comput.-Hum. Interact., 10 (2), 119-149.

Hornbæk, K. and Hertzum, M. (2007). Untangling the usability of fisheye menus. ACM Trans. Comput.-Hum. Interact., 14 (2), 6.

Kim, J. et al. (2000). How Do We Understand a System with (So) Many Diagrams? Cognitive Integration Processes in Diagrammatic Reasoning. Information Systems Research, 11 (3), 284-303.

Kobsa, A. (2004). User Experiments with Tree Visualization Systems. In Proceedings of the Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, p. 9-16.

Koschmider, A. and Blanchard, E. (2007). User Assistance for Business Process Model Decomposition. In Proceedings of the First IEEE International Conference on Research Challenges in Information Science, p. 445-454, Ouarzazate, Marokko.

Koschmider, A. et al. (2013). Analysis of Process Model Reuse: Where Are We Now, Where Should We Go From Here? Working Paper. Karlsruhe Institute of Technology.

Lindsay, A. et al. (2003). Business processes—attempts to find a definition. Information and Software Technology, 45 (15), 1015-1019.

Maes, A. and Poels, G. (2007). Evaluating Quality of Conceptual Modelling Scripts Based on User Perceptions. Data & Knowledge Engineering, 63 (3), 701-724.

Malinova, M. et al. (2013). An Empirical Investigation on the Design of Process Architectures. In Proceedings of the 11th International Conference on Wirtschaftsinformatik, Leipzig, Germany.

Malone, T.W. et al. (eds.) 2003. Organizing Business Knowledge. The MIT Process Handbook, The MIT Press.

Mendling, J. et al. (2010). Seven process modeling guidelines (7PMG). Information and Software Technology, 52 (2).

Moody, D. (2006a). Dealing with "Map Shock": A Systematic Approach for Managing Complexity in Requirements Modelling. In Proceedings of the Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), Luxembourg.

Moody, D.L. (2006b). What Makes a Good Diagram? Improving the Cognitive Effectiveness of Diagrams in IS Development. In Proceedings of the 15th International Conference on Information Systems Development (ISD 2006), Budapest, Hungary.

Moody, D.L. (2009). The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering, 35 (5), 756-779.

North, C. (2005). Information Visualization. In Salvendy, G. (ed.) Handbook of Human Factors and Ergonomics. John Wiley & Sons, New York.

Nunnally, J.C. and Bernstein, I.H. (1994). Psychometric Theory. McGraw-Hill, New York, New York.

Parnas, D.L. (1972). On the criteria to be used in decomposing systems into modules. Commun. ACM, 15 (12), 1053-1058.

Paulson, D. and Wand, Y. (1992). An Automated Approach to Information Systems Decomposition. IEEE Trans. Softw. Eng., 18 (3), 174-189.

Polyvyanyy, A. et al. (2008). Process Model Abstraction: A Slider Approach. In Proceedings of the Proceedings of the 2008 12th International IEEE Enterprise Distributed Object Computing Conference, p. 325-331.

Polyvyanyy, A. et al. (2009). On Application of Structural Decomposition for Process Model Abstraction. In Proceedings of the 2nd International Conference on Business Process and Services Computing, Leipzig, Germany.

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph., 11 (1), 92-99.

Sinur, J. and Hill, J.B. (2010). Magic Quadrant for Business Process Management Suites. Gartner RAS Core research note, 1-24.

Smirnov, S. et al. (2012). Business process model abstraction: a definition, catalog, and survey. Distrib. Parallel Databases, 30 (1), 63-99.

Storey, M.-A. and Müller, H. (1996). Graph layout adjustment strategies. In Brandenburg, F. (ed.) Graph Drawing. Springer Berlin Heidelberg.

Summers, K.L. et al. (2003). An experimental evaluation of continuous semantic zooming in program visualization. In Proceedings of the Proceedings of the Ninth annual IEEE conference on Information visualization, p. 155-162, Seattle, Washington.

Sweller, J. and Chandler, P. (1994). Why Some Material Is Difficult to Learn. Cognition and Instruction, 12 (3), 185-233.

Wand, Y. and Weber, R. (1995). On the deep structure of information systems. Information Systems Journal, 5 (3), 203-223.

Ware, C. (2004). Information Visualization. Elsevier. Morgan Kaufmann, San Francisco.

Weidlich, M. et al. (2010). Efficient Consistency Measurement Based on Behavioural Profiles of Process Models. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 99 (PrePrints).

Woods, D.D. (1984). Visual momentum: a concept to improve the cognitive coupling of person and computer. Int. J. Man-Mach. Stud., 21 (3), 229-244.

Zugal, S. et al. (2012). Assessing the Impact of Hierarchy on Model - A Cognitive Perspective. In Proceedings of the EESSMod.